731500

BUDDHA, COMPARE, AND OTHER EMP ENVIRONMENT DATA REDUCTION CODES

JUNE 1971

U. S. ARMY HARRY DIAMOND LABORATORIES
WASHINGTON, D. C.

PREPARED BY:

David L. Jones
Deborah H. Stump

BRADDOCK, DUNN AND McDONALD, INC.
WASHINGTON, D.C. OFFICE

UNDER CONTRACT NUMBER DAAK02-70-C-0258

This project has been funded by the
Defense Atomic Support Agency
under Nuclear Weapons Effects
Research Subtask Numbers
EA-091 and EA-094

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Braddock, Dunn and McDonald, Inc. 8027 Leesburg Pike McLean, Virginia 22101	Unclassified
	2b. GROUP

3. REPORT TITLE

BUDDHA, COMPARE, AND OTHER EMP ENVIRONMENT DATA REDUCTION CODES

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

FINAL REPORT Jan 1971 - June 1971

5. AUTHOR(S) (First name, middle initial, last name)

David L. Jones and Deborah H. Stump

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1971	203	2

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
DAAK02-70-C-0258	
b. PROJECT NO.	BDM/W-TR71-66-0258
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U.S. Army Harry Diamond Laboratories Washington, D.C. 20438

13. ABSTRACT

The programs documented in this report are part of a comprehensive package of digital computer codes which calculate the electromagnetic pulse (EMP) environment that is produced by a nuclear explosion. Each new program is described in the form of a user's guide which includes a discussion on how the program operates, instructions on how to execute the program, and sample output generated by the program. For each modified program the discussion tells how and why it was modified.

DD FORM 1 NOV 68 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Electromagnetic Pulse (EMP)						
Digital Computer Code						
Graphical Display of Output						

SUMMARY

This is the final report for contract DAAK02-70-C-0258 and covers the period from January 1971 through June 1971. It is a summary of all programs written and/or modified under this contract during that period by Braddock, Dunn and McDonald, Inc. for the U.S. Army Mobility Equipment Research and Development Center and subsequently, for U.S. Army Harry Diamond Laboratories. Each new program is described in the form of a user's guide which includes a discussion on how the program operates, instructions on how to execute the program, and sample output generated by the program. For each modified program the discussion tells how and why it was modified.

FOREWORD

The programs documented in this report are part of a comprehensive package of digital computer codes which calculate the electromagnetic pulse (EMP) environment that is produced by a nuclear explosion. This work is a continuation of an extensive technical effort formerly conducted at the U.S. Army Mobility Equipment Research and Development Center (MERDC), Fort Belvoir, Virginia, and presently conducted at the U.S. Army Harry Diamond Laboratories, Washington, D.C. The modification of previously documented computer codes and the writing of new codes has necessitated this documentation.

The programming tasks were performed under contract DAAK02-70-C-0258 by Mr. Jeffrey A. Borbely (project leader), Mr. David L. Jones, and Mrs. Deborah H. Stump of Braddock, Dunn and McDonald, Inc. Technical supervisor was Mr. William T. Wyatt, Jr. formerly of the Physics Division, Electromagnetic Effects Laboratory, MERDC, and presently of Branch 1030, HDL.

TABLE OF CONTENTS

		Page
I.	NEW CODES AND MAJOR MODIFICATIONS	
A.	COMPARE	1
1.	Introduction	1
2.	Plot Definition and Control	
3.	Subroutines LINEUP and LYNEUP	3
4.	Subroutines LTCONE and LTKONE	4
5.	Subroutines TIMAXS and TYMAXS	4
6.	Subroutine COEF	5
7.	Subroutine SCALE	6
8.	Subroutine LINES	6
9.	Subroutines GREEK, SEPAR8, and ISHIFT	6
10.	Description of Input	7
a.	Type 1	7
b.	Type 2	9
c.	Type 3	12
d.	Type 4	14
e.	Type 5	17
11.	Sample Output	21
12.	Program Listing	41
B.	BUDDHA	77
1.	Introduction	77
2.	Input Revisions	77
3.	Data Revisions	81
4.	Plot Additions	81
5.	Subroutine LINES	83
6.	Subroutine LOGPLT	83
7.	CALCOMP Subroutines	84
8.	Sample Output	85
9.	Program Listing	89
C.	GREEK	115
1.	Introduction	115
2.	Calling Procedure	115
3.	Data Control	116
4.	Subroutine SEPAR8	116
5.	Function Subroutine ISHIFT	116
6.	Special Considerations	117
7.	Sample Output	118
8.	Program Listing	120

TABLE OF CONTENTS (continued)

		Page
II.	MINOR MODIFICATIONS	
	A. ELECTRA	125
	1. Description of Modifications	125
	2. Program Listing	127
	B. ORESTES	153
	1. Description of Modifications	153
	2. Program Listing	154
	C. REDACT	181
	1. Description of Modifications	181
	2. Program Listing	182
	D. REDACTO	191
	1. Description of Modifications	191
	2. Program Listing	192

I. NEW CODES AND MAJOR MODIFICATIONS

A. COMPARE

1. Introduction

Program COMPARE is a plot program which draws from one to ten parallel curves on the same plotting area to facilitate data comparison. Given a set of EMP field values which are a function of three independent parameters (range, time, and theta), the user may elect to hold any one of the three parameters constant, have each of the ten curves represent a different value of the second parameter, and plot the EMP field values versus the remaining parameter. Each of the curves may be scaled independently or the data for all the curves may be scaled as one set.

2. Plot Definition and Control

The EMP field data is supplied to program COMPARE by reading the output tapes from program ELECTRA the ground burst EMP code.[1,2] In order to determine which data is to be read from the tape a series of input data cards are read by program COMPARE. The detailed structure of these cards

---

1. Jones, D. L. and D. H. Stump, ELECTRA, ORESTES, and Supporting Graphic Display Codes, U.S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Virginia, February 1971, pp. 1-36.

2. Borbely, J.A. and D. L. Jones, ELECTRA, An Electromagnetic Pulse Fortran Program (User's Guide), U.S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Virginia, October 1969.

will be discussed in Section I.A.10. The first data card determines which type of plot is desired. That is, it establishes which parameter will be held constant, which parameter (designated z-axis) will change only from curve to curve, and which parameter (designated x-axis) will be plotted against the field data. This first card will contain one of the following ten-character keywords:

PLOT TYPE	KEY WORD	Z-AXIS	X-AXIS	CONSTANT
1	TIME-THETA	TIME	THETA	RANGE
2	RANGE-THETA	RANGE	THETA	TIME
3	TIME-RANGE	TIME	RANGE	THETA
4	RANGE-TIME	RANGE	TIME	THETA
5	THETA-TIME	THETA	TIME	RANGE

These five key words are used as switches to control the execution of five separate sections of program COMPARE. The THETA-RANGE-TIME combination is not implemented in this version of COMPARE.

After the program enters one of the five sections, the additional data items needed to set up that type of plot are read from cards. These data items include the value of the constant parameter, the values of the z-axis parameters, and the values or limits of the x-axis parameters. In addition controls for scaling and for x-axis spacing are also set by data card. Once these parameters have been set, the EMP data is read from tape, edited and ordered accordingly, and then transferred to the appropriate plotting subroutine (LINEUP, LYNEUP, LTCONE; LTKONE, TIMAXS, or TYMAXS).

3. Subroutines LINEUP and LYNEUP

Subroutine LINEUP is called by program COMPARE to produce plots of types 1 and 2 as defined in Section 1.A.2. It calls subroutine SCALE, treating the EMP field data for all curves as one set, and receives one maximum, one minimum, and one delta y scale factor. LINEUP then draws each curve separately, scaling each data set against the maximum, the minimum, and the delta y scale factor. Each curve has its own set of axes appropriately annotated.

If the plot is of type 1, the constant parameter is range and is annotated by the statement ALL CURVES ARE FOR R = (range). Each curve represents a different value of either source time or retarded time and is labeled either T = (source time) or τ = (retarded time). The x-axis shows the values of theta which may be spaced according to actual value or spaced at equal intervals as a user option.

If the plot is of type 2, the constant parameter is either source time or retarded time and is annotated either by the statement ALL CURVES ARE FOR T = (source time) or by ALL CURVES ARE FOR τ = (retarded time). Each curve represents a different value of range and is labeled R = (range). The x-axis again shows the values of theta as described above.

Subroutine LYNEUP differs from LINEUP only in the area of data scaling. In subroutine LYNEUP the EMP field data set for each curve is supplied to subroutine SCALE separately. The resulting maximum, minimum, and scale factor for each curve are then reflected in the annotation on its set of axes.

4. Subroutines LTCONE and LTKONE

Subroutine LTCONE is called by program COMPARE to produce plots of type 3 as defined in Section I.A. 2. In a manner similar to that described for subroutine LINEUP in Section I.A.3., subroutine LTCONE calls subroutine SCALE, treating the EMP field data for all curves as one set, and then draws the curves separately with individually annotated axes.

For a type 3 plot the constant parameter is theta and is annotated by the statement ALL CURVES ARE FOR θ = (theta). Each curve represents a different value of source time and is labeled T = (source time). The x-axis shows values of retarded range R' where $R'=R-cxT$, R = actual range, T = source time, and c = speed of light. Since each curve represents a different value of T, different retarded ranges must be calculated for each curve and then each curve must be shifted to align with the annotated x-axis at the bottom of the plot.

Subroutine LTKONE differs from LTCONE only in the area of data scaling. In subroutine LTKONE the EMP field data set for each curve is supplied to subroutine SCALE separately. The resulting maximum, minimum, and scale factor for each curve are then reflected in the annotation on its set of axes.

5. Subroutines TIMAXS and TYMAXS

Subroutines TIMAXS and TYMAXS are called by program COMPARE to produce plots of types 4 and 5 as defined in Section I.A.2. Subroutine

TIMAXS calls subroutine SCALE, treating the EMP field data for all curves as one set, and receives one maximum, one minimum, and one delta y scaling factor. TIMAXS then draws each curve separately, scaling the data against the maximum, minimum, and scale factor established for the total data set. Each curve has its own set of axes appropriately annotated.

Subroutine TYMAXS supplies subroutine SCALE with the data for each curve separately and then plots each curve according to its own maximum, minimum, and delta y scale factor. Each curve again has its own set of axes appropriately annotated.

If the plot is of type 4, the constant parameter is theta and is annotated by the statement ALL CURVES ARE FOR θ = (theta). Each curve represents a different value of range and is labeled R = (range). The x-axis shows the values of either source time or retarded time in units of nanoseconds and is labeled either T IN NSEC or τ IN NSEC. The time values may be spaced according to actual value or at equal intervals as a user option.

If the plot is of type 5, the constant parameter is range and is annotated by the statement ALL CURVES ARE FOR R = (range). Each curve represents a different value of theta and is labeled θ = (theta). The x-axis is the same as described above for type 4 plots.

6. Subroutine COEF

Subroutine COEF helps set up the y-axis labeling for the six plotting subroutines described in Sections I.A.3., I.A.4., and I.A.5.

Given a delta y scale factor, subroutine COEF breaks the factor apart into a coefficient between one and ten and a power of ten. This coefficient is then used to determine the number of intervals into which the y-axis will be divided. The power of ten is used along with the minimum y value to determine the annotation to be written beside each tic mark on the y-axis.

7. Subroutine SCALE

Subroutine SCALE performs data scaling for the six plotting subroutines described in Sections I.A.3., I.A.4., and I.A.5. Given a data array, subroutine SCALE scans the array to find the maximum and minimum values. It then defines the delta y scale factor by the following expression:

Δy = (maximum y - minimum y)/ length of plot axis

8. Subroutine LINES

Subroutine LINES is a plot routine designed for drawing data curves. Given two data arrays which are to be plotted against each other, subroutine LINES performs a curve fit to the data in a manner such that the curve can be represented by a series of equally spaced dots, line segments or both.

9. Subroutines GREEK, SEPAR8, and ISHIFT

These three subroutines work together to draw the letters of the Greek alphabet in either upper case or lower case form. They will be described in detail in Section I.C. of this report. They are called by program COMPARE to supply τ's and θ's for labeling the plots.

10. Description of Input

The card input for program COMPARE can best be described by breaking it up into five distinct groups even though several of the groups will have the same card types.

a. Type 1

The input for plots of type 1 as defined in Section I.A.2. consists of the five card types (1A, 1B, 1C, 1D, and 1E) shown below:

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
1A	1-10	Select type of plot	CHOICE	Alpha-betic	Must equal TIME-THETA
	11-80	Not used			Blank
1B	1-10	Number of curves	NZ	Inte-ger	Determines number of time values to appear on card 1C below. Must be $\leq$ 10 in present version
	11-80	Not used			Blank
1C	1-10	Time value for 1st curve	Z(1)	Float-ing	May be source time or retarded time in seconds (See TCOL below)
	11-20	Time value for 2nd curve	Z(2)	Float-ing	May be source time or retarded time in seconds (See TCOL below)
	21-30	Time value for 3rd curve	Z(3)	Float-ing	May be source time or retarded time in seconds (See TCOL below)
	31-40	Time value for 4th curve	Z(4)	Float-ing	May be source time or retarded time in seconds (See TCOL below)
	41-50	Time value for 5th curve	Z(5)	Float-ing	May be source time or retarded time in seconds (See TCOL below)

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
1C cont.	51-60	Time value for 6th curve	Z(6)	Floating	May be source time or retarded time in seconds (See TCOL below)
	61-70	Time value for 7th curve	Z(7)	Floating	May be source time or retarded time in seconds (See TCOL below)
	71-80	Time value for 8th curve	Z(8)	Floating	May be source time or retarded time in seconds (See TCOL below)
1D	1-10	Time increment	DT	Floating	Used as tolerance factor in matching time values
	11-20	Range increment	DR	Floating	Used as tolerance factor in matching range values
	21-30	Constant parameter	CONVAR	Floating	Range value for all curves
	31-40	Time indicator	TCOL	Floating	TCOL=0, retarded time TCOL≠0, source time
	41-50	Length of x-axis	XLEN	Floating	6 inches recommended
	51-60	Length of y-axis	YLEN	Floating	This is the combined total length of all the y-axes. Each individual axis length will equal: (YLEN/NZ) - .25 YLEN must be $\leq$ 10.
	61-80	Not used			Blank
1E	1-4	EMP field to be plotted	AVAR	Alphabetic	Must equal BPHI, ETHE, ERAD, or SIGT.
	5-10	Not used			Blank
	11-20	Minimum θ value	THETMN	Floating	Minimum theta value for x-axis data.

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
1E cont.	21-30	Maximum θ value	THETMX	Floating	Maximum theta value for x-axis data.
	31-40	x-axis spacing control	INCX	Integer	INCS=1, spaced according to data; INCX=2, equally spaced
	41-50	y-axis scaling control	ISCALE	Integer	ISCALE=0, scale all curves as one set; ISCALE ≠0, scale each curve separately.
	51-80	Not used			Blank

A standard deck for type 1 plots will consist of one type 1A, one type 1B, one type 1C (a second type 1C is required if NZ is greater than eight), one type 1D, and as many type 1E cards as desired. The last type 1E card must be followed by a blank card which signals the end of that set of type 1 plots.

b. Type 2

The input for plots of type 2 as defined in Section I.A.2. consists of the five card types (2A, 2B, 2C, 2D, 2E) shown below:

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
2A	1-10	Select type of plot	CHOICE	Alphabetic	Must equal RANG-THETA
	11-80	Not used			Blank
2B	1-10	Number of curves	NZ	Integer	Determines number of range values to appear on card 2C below. Must be ≤ 10 in present version.

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
2B cont.	11-80	Not used			Blank
2C	1-10	Range value for 1st curve	Z(1)	Floating	Range in meters
	11-20	Range value for 2nd curve	Z(2)	Floating	Range in meters
	21-30	Range value for 3-d curve	Z(3)	Floating	Range in meters
	31-40	Range value for 4th curve	Z(4)	Floating	Range in meters
	41-50	Range value for 5th curve	Z(5)	Floating	Range in meters
	51-60	Range value for 6th curve	Z(6)	Floating	Range in meters
	61-70	Range value for 7th curve	Z(7)	Floating	Range in meters
	71-80	Range value for 8th curve	Z(8)	Floating	Range in meters
2D	1-10	Time increment	DT	Floating	Used as tolerance factor in matching time values
	11-20	Range increment	DR	Floating	Used as tolerance factor in matching range values
	21-30	Constant parameter	CONVAR	Floating	Time value for all curves. May be source time or retarded time
	31-40	Time indicator	TCOL	Floating	TCOL=0, retarded time TCOL≠0, source time
	41-50	Length of x-axis	XLEN	Floating	6 inches recommended

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
2D cont.	51-60	Length of y-axis	YLEN	Floating	This is the combined total length of all the y-axes. Each individual axis length will equal: (YLEN/NZ) -.25 YLEN must be $\leq$ 10.
	61-80	Not used			Blank
2E	1-4	EMP field to be plotted	AVAR	Alphabetic	Must equal BPHI, ETHE, ERAD, or SIGT.
	5-10	Not used			Blank
	11-20	Minimum θ value	THETMN	Floating	Minimum theta value for x-axis data.
	21-30	Maximum θ value	THETMX	Floating	Maximum theta value for x-axis data.
	31-40	x-axis spacing control	INCX	Integer	INCX=1, spaced according to data; INCX=2, equally spaced
	41-50	y-axis scaling control	ISCALE	Integer	ISCALE=0, scale all curves as one set; ISCALE $\neq$ 0, scale each curve separately.
	51-80	Not used			Blank

A standard deck for type 2 plots will consist of one type 2A, one type 2B, one type 2C (a second type 2C is required if NZ is greater than eight), one type 2D, and as many type 2E cards as desired. The last type 2E card must be followed by a blank card which signals the end of that set of type 2 plots.

c. Type 3

The input for plots of type 3 as defined in Section I.A.2. consists of the six card types (3A, 3B, 3C, 3D, 3E, and 3F) shown below:

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
3A	1-10	Select type of plot	CHOICE	Alpha-betic	Must equal TIME-RANGE
	11-80	Not used			Blank
3B	1-10	Number of curves	NZ	Inte-ger	Determines number of time values to appear on card 3C below. Must be $\leq$ 10 in present version.
	11-80	Not used			Blank
3C	1-10	Time value for 1st curve	Z(1)	Float-ing	Source time value in seconds
	11-20	Time value for 2nd curve	Z(2)	Float-ing	Source time value in seconds
	21-30	Time value for 3rd curve	Z(3)	Float-ing	Source time value in seconds
	31-40	Time value for 4th curve	Z(4)	Float-ing	Source time value in seconds
	41-50	Time value for 5th curve	Z(5)	Float-ing	Source time value in seconds
	51-60	Time value for 6th curve	Z(6)	Float-ing	Source time value in seconds
	61-70	Time value for 7th curve	Z(7)	Float-ing	Source time value in seconds
	71-80	Time value for 8th curve	Z(8)	Float-ing	Source time value in seconds

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
3D	1-10	Time increment	DT	Floating	Used as tolerance factor in matching time values
	11-20	Range increment	DR	Floating	Used as tolerance factor in matching range values
	21-30	Length of x-axis	XLEN	Floating	6 inches recommended
	31-40	Length of y-axis	YLEN	Floating	This is the combined total length of all the y-axes. Each individual axis length will equal (YLEN/NZ) - .25 YLEN must be $\leq$ 10.
	41-80	Not used			Blank
3E	1-10	Number of range values	NRANGE	Integer	Number of range values to be calculated for X-axis.
	11-20	First range value	RMIN	Floating	First range value, basis for calculating all other range values
	21-30	Increment between range values	RINC	Floating	Each range value will equal the previous value plus RINC
	31-80	Not used			Blank
3F	1-4	EMP field to be plotted	AVAR	Alphabetic	Must equal BPHI, ETHE, ERAD, or SIGT
	5-10	Not used			Blank
	11-20	Constant parameter	CONVAR	Floating	Theta value for all curves
	21-30	X-axis spacing control	INCX	Integer	INCX=1, spaced according to data; INCX=2, equally spaced

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
3F cont.	31-40	y-axis scaling control	ISCALE	Integer	ISCALE=0, scale all curves as one set; ISCALE ≠ 0, scale each curve separately.
	41-80	Not used			Blank

A standard deck for type 3 plots will consist of one type 3A, one type 3B, one type 3C (a second type 3C is required if NZ is greater than eight), one type 3D, one type 3E, and as many type 3F cards as desired. The last type 3F card must be followed by a blank card which signals the end of that set of type 3 plots.

d. Type 4

The input for plots of type 4 as defined in Section I.A.2. consists of the seven card types (4A, 4B, 4C, 4D, 4E, 4F, and 4G) shown below:

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
4A	1-10	Select type of plot	CHOICE	Alphabetic	Must equal RANGE-TIME
	11-80	Not used			Blank
4B	1-10	Number of curves	NZ	Integer	Determines number of range values to appear on card 4C below. Must be ≤ 10 in present version.
	11-80	Not used			Blank

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
4C	1-10	Range value for 1st curve	Z(1)	Floating	Range in meters
	11-20	Range value for 2nd curve	Z(2)	Floating	Range in meters
	21-30	Range value for 3rd curve	Z(3)	Floating	Range in meters
	31-40	Range value for 4th curve	Z(4)	Floating	Range in meters
	41-50	Range value for 5th curve	Z(5)	Floating	Range in meters
	51-60	Range value for 6th curve	Z(6)	Floating	Range in meters
	61-70	Range value for 7th curve	Z(7)	Floating	Range in meters
	71-80	Range value for 8th curve	Z(8)	Floating	Range in meters
4D	1-10	Time increment	DT	Floating	Used as tolerance factor in matching time values
	11-20	Range increment	DR	Floating	Used as tolerance factor in matching range values
	21-30	Length of x-axis	XLEN	Floating	6 inches recommended
	31-40	Length of y-axis	YLEN	Floating	This is the combined total length of all the y-axes. Each individual axis length will equal (YLEN/NZ) -.25 YLEN must be $\leq$ 10.
	41-80	Not used			Blank.

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
4E	1-10	Number of different DELT's	NDELTS	Integer	This controls the number of type 4F cards to be read.
	11-20	Minimum time value	TMIN	Floating	First time value, basis for calculating all other time values. May be source or retarded time (See TCOL below).
	21-80	Not used			Blank
4F	1-10	Number of times to use the DELT value found on this card	NDELT	Integer	Each time value for the x-axis is equal to the previous value plus DELT. NDELT tells how many times to add DELT.
	11-20	Increment between time values.	DELT	Floating	See description of NDELT above.
	21-80	Not used			Blank
4G	1-4	EMP field to be plotted	AVAR	Alphabetic	Must equal BPHI, ETHE, ERAD, or SIGT.
	5-10	Not used			Blank
	11-20	Constant parameter	CONVAR	Floating	Theta value for all curves
	21-30	x-axis spacing control	INCX	Integer	INCX=1, spaced according to data; INCS=2, equally spaced.
	31-40	y-axis scaling control	ISCALE	Integer	ISCALE=0, scale all curves as one set; ISCALE ≠ 0, scale each curve separately.
	41-50	Time indicator	TCOL	Floating	TCOL=0, retarded time TCOL≠0, source time
	51-80	Not used			Blank

A standard deck for type 4 plots will consist of one type 4A, one type 4B, one type 4C (a second type 4C is required if NZ is greater than eight), one type 4D, one type 4E, as many type 4F as specified by the NDELTS parameter, and as many type 4G cards as desired. The last type 4G card must be followed by a blank card which signals the end of that set of type 4 plots.

e. Type 5

The input of plots of type 5 as defined in Section I.A.2. consists of the seven card types (5A, 5B, 5C, 5D, 5E, 5F, and 5G) shown below:

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
5A	1-10	Select type of plot	CHOICE	Alpha-betic	Must equal THETA-TIME
	11-80	Not used			Blank
5B	1-10	Time increment	DT	Float-ing	Used as tolerance factor in matching time values
	11-20	Range increment	DR	Float-ing	Used as tolerance factor in matching range values
	21-30	Length of x-axis	XLEN	Flota-ing	6 inches recommended
	31-40	Length of y-axis	YLEN	Float-ing	This is the combined total length of all the y-axes. Each individual axis length will equal (YLEN/NZ) -.25 YLEN must be $\leq$ 10.
	41-50	Constant parameter	CONVAR	Float-ing	Range value for all curves.

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
5B cont.	51-80	Not used			Blank
5C	1-10	Number of different DELT's	NDELTS	Integer	This controls the number of type 5D cards to be read
	11-20	Minimum time value	TMIN	Floating	First time value, basis for calculating all other time values. May be source or retarded time (see TCOL below).
	21-80	Not used			Blank
5D	1-10	Number of times to use the DELT value found on this card	NDELT	Integer	Each time value for the x-axis is equal to the previous value plus DELT. NDELT tells how many times to add DELT.
	11-20	Increment between time values.	DELT	Floating	See description of NDELT above.
	21-80	Not used			Blank
5E	1-10	Number of curves	NZ	Integer	Determines number of theta values to appear on card 5F below. Must be $\leq$ 10 in present version.
	11-80	Not used			Blank
5F	1-10	Theta value for 1st curve	Z(1)	Floating	Theta value in degrees
	11-20	Theta value for 2nd curve	Z(2)	Floating	Theta value in degrees
	21-30	Theta value for 3rd curve	Z(3)	Floating	Theta value in degrees

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENTS	PROGRAM NAME	TYPE	DESCRIPTION
5F cont.	31-40	Theta value for 4th curve	Z(4)	Floating	Theta value in degrees
	41-50	Theta value for 5th curve	Z(5)	Floating	Theta value in degrees
	51-60	Theta value for 6th curve	Z(6)	Floating	Theta value in degrees
	61-70	Theta value for 7th curve	Z(7)	Floating	Theta value in degrees
	71-80	Theta value for 8th curve	Z(8)	Floating	Theta value in degrees
5G	1-4	EMP field to be plotted	AVAR	Alphabetic	Must equal BPHI, ETHE, ERAD, or SIGT
	5-10	Not used			Blank
	11-20	Time indicator	TCOL	Floating	TCOL=0, retarded time TCOL≠0, source time
	21-30	x-axis spacing control and theta redefinition control	INCX	Integer	INCX=1, spaced according to data; INCX=2, equally spaced. If INCX=3, a new set of 5E-5F cards are read to redefine the theta values.
	31-40	y-axis scaling control	ISCALE	Integer	ISCALE=0, scale all curves as one set; ISCALE≠ 0, scale each curve separately
	41-80	Not used			Blank

A standard deck for type 5 plots will consist of one type 5A, one type 5B, one type 5C, as many type 5D as specified by the NDELTS parameter, one type 5E, one type 5F (a second type 5F is required if NZ is greater than eight), and as many type 5G cards as desired. If the last type 5G card sets

INCX=3, then the series 5E-5G may be repeated. The last type 5G card must be followed by a blank card which signals the end of that set of type 5 plots.

Plots of types 1 through 5 may be requested in any order desired as long as all the cards needed for a particular type are grouped together. To stop execution of the plot cycle and close the plot file, a card with the word STOP in columns 1-4 must be placed after the last blank card.

11. Sample Output

Description of Sample Output

Figure	Plot Type	Time Values	Y-axis Scaling	X-axis Spacing
1	1	Retarded	As one set	By value
2	1	Retarded	As one set	Equally spaced
3	1	Source	As one set	Equally spaced
4	1	Source	Separately	Equally spaced
5	2	Retarded	As one set	By value
6	2	Retarded	As one set	Equally spaced
7	2	Source	As one set	Equally spaced
8	2	Source	Separately	Equally spaced
9	3	Source	Separately	By actual value*
10	3	Source	Separately	By retarded value
11	4	Retarded	As one set	By value
12	4	Retarded	Separately	By value
13	4	Source	As one set	By value
14	4	Source	Separately	By value
15	5	Retarded	As one set	By value
16	5	Retarded	Separately	By value
17	5	Source	As one set	By value
18	5	Source	Separately	By value

*Note: The actual value data points should have the same relative spacing as the retarded value data points. The following modification to program COMPARE should eliminate the discrepancy: The first statement following statement 1220 should read X(I) = I*RINC.

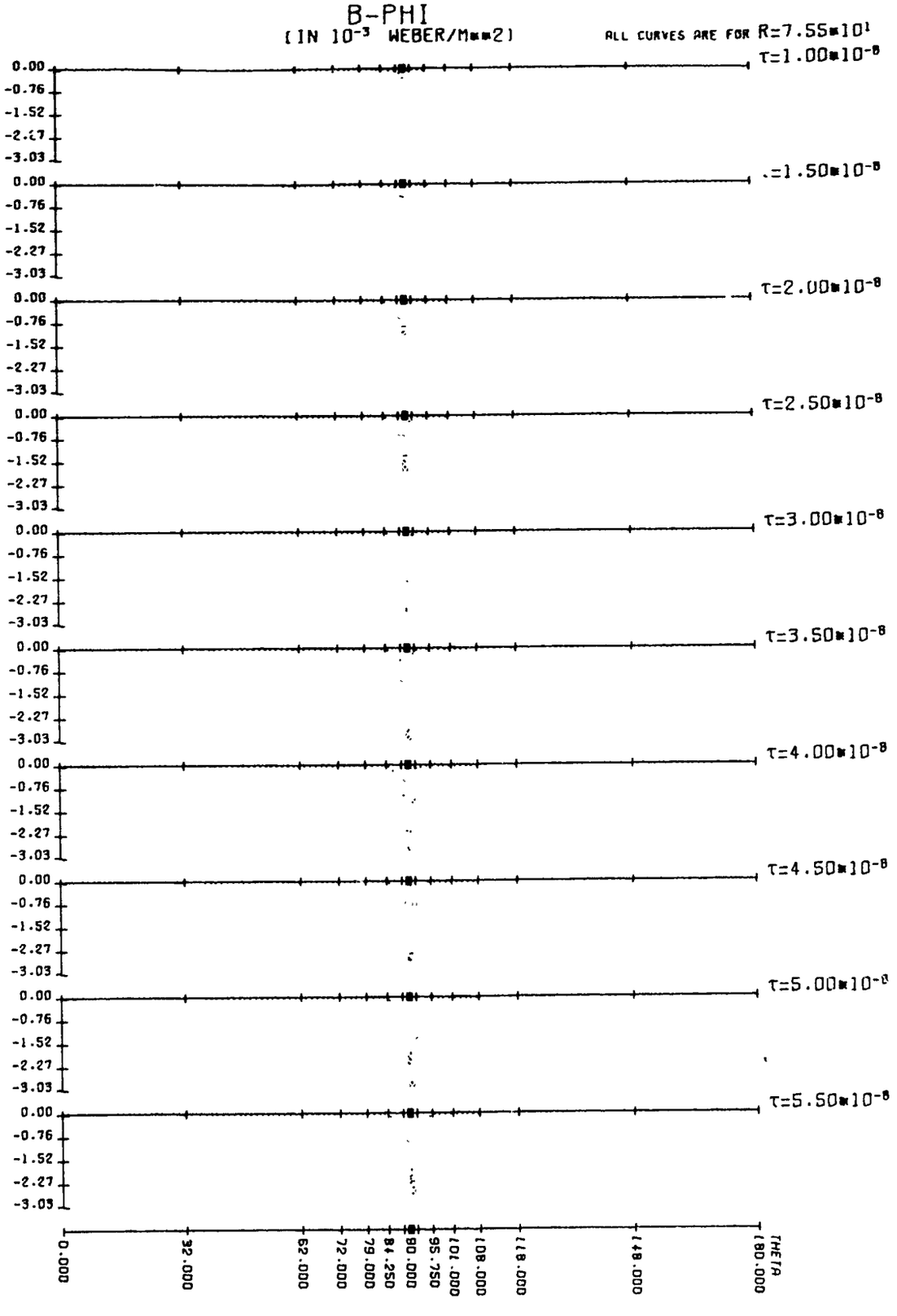

B-PHI
(IN 10⁻³ WEBER/M**2)
ALL CURVES ARE FOR R=7.55*10¹
τ=1.00*10⁻⁸
τ=1.50*10⁻⁸
τ=2.00*10⁻⁸
τ=2.50*10⁻⁸
τ=3.00*10⁻⁸
τ=3.50*10⁻⁸
τ=4.00*10⁻⁸
τ=4.50*10⁻⁸
τ=5.00*10⁻⁸
τ=5.50*10⁻⁸
0.00
-0.76
-1.52
-2.27
-3.03
0.000
32.000
63.000
72.000
76.000
84.250
90.000
95.750
101.000
108.000
118.000
148.000
180.000
THETA

Figure 1.

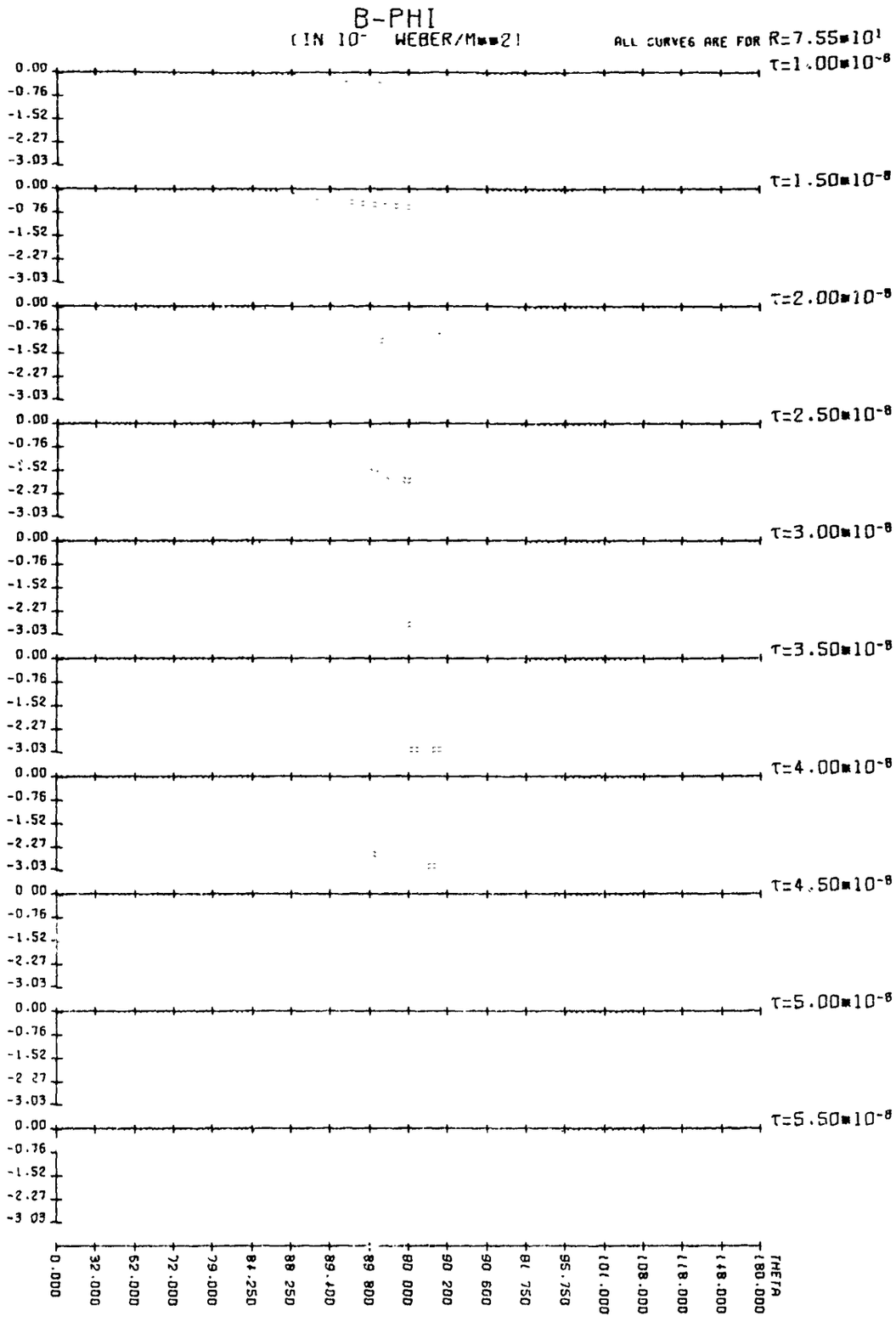

B-PHI
(IN 10⁻ WEBER/M**2)
ALL CURVES ARE FOR R=7.55*10¹
τ=1.00*10⁻⁸
τ=1.50*10⁻⁸
τ=2.00*10⁻⁸
τ=2.50*10⁻⁸
τ=3.00*10⁻⁸
τ=3.50*10⁻⁸
τ=4.00*10⁻⁸
τ=4.50*10⁻⁸
τ=5.00*10⁻⁸
τ=5.50*10⁻⁸
0.00
-0.76
-1.52
-2.27
-3.03
0.000
32.000
52.000
72.000
79.000
84.250
88.250
89.400
89.800
90.000
90.200
90.600
91.750
95.750
101.000
108.000
118.000
148.000
180.000
THETA

Figure 2.

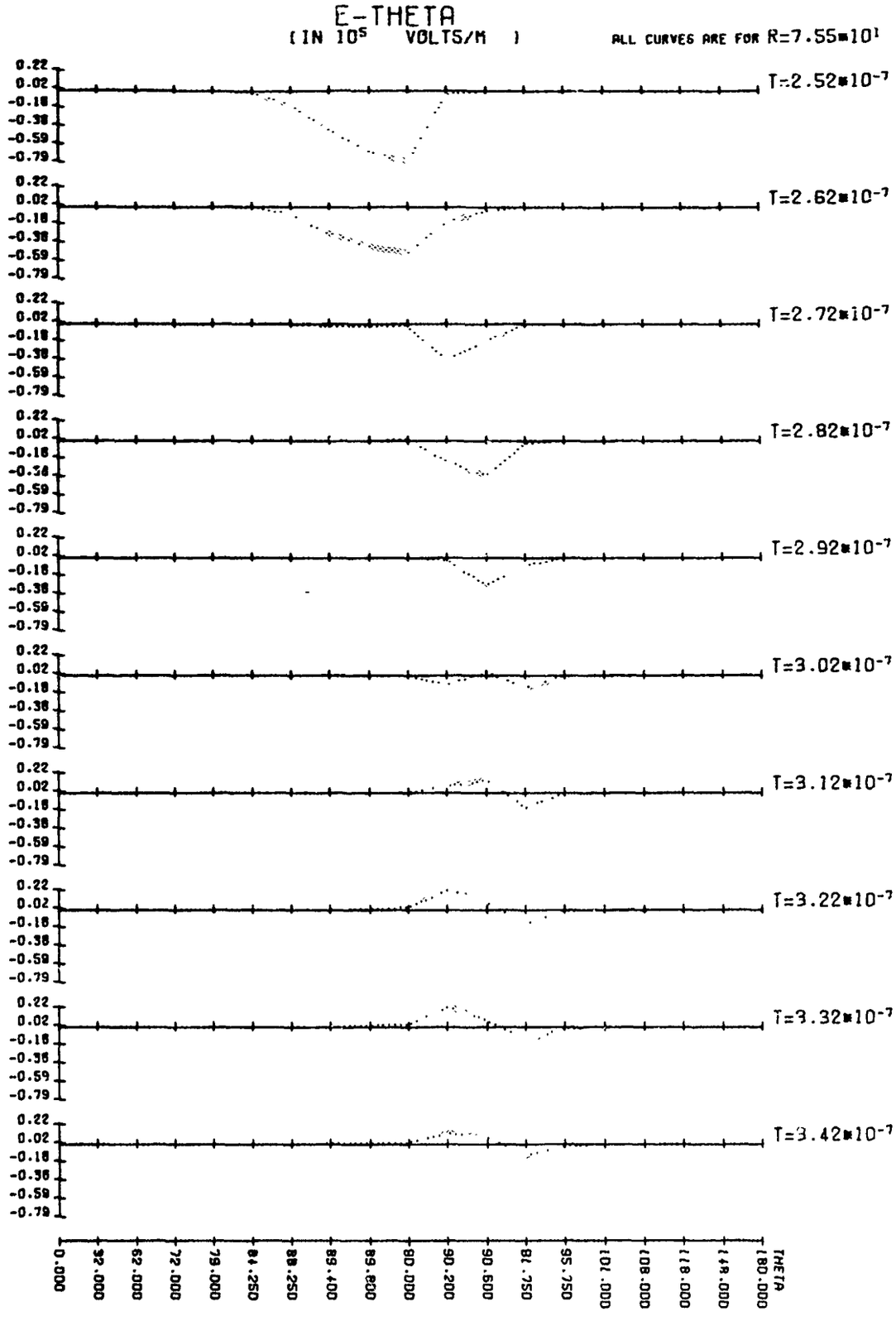

E-THETA
(IN 10^5 VOLTS/M)
ALL CURVES ARE FOR R=7.55*10^1
T=2.52*10^-7
T=2.62*10^-7
T=2.72*10^-7
T=2.82*10^-7
T=2.92*10^-7
T=3.02*10^-7
T=3.12*10^-7
T=3.22*10^-7
T=3.32*10^-7
T=3.42*10^-7
0.22
0.02
-0.18
-0.38
-0.59
-0.79
0.000
12.000
24.000
36.000
48.000
60.000
72.000
84.000
96.000
108.000
120.000
132.000
144.000
156.000
168.000
180.000
THETA

Figure 3.

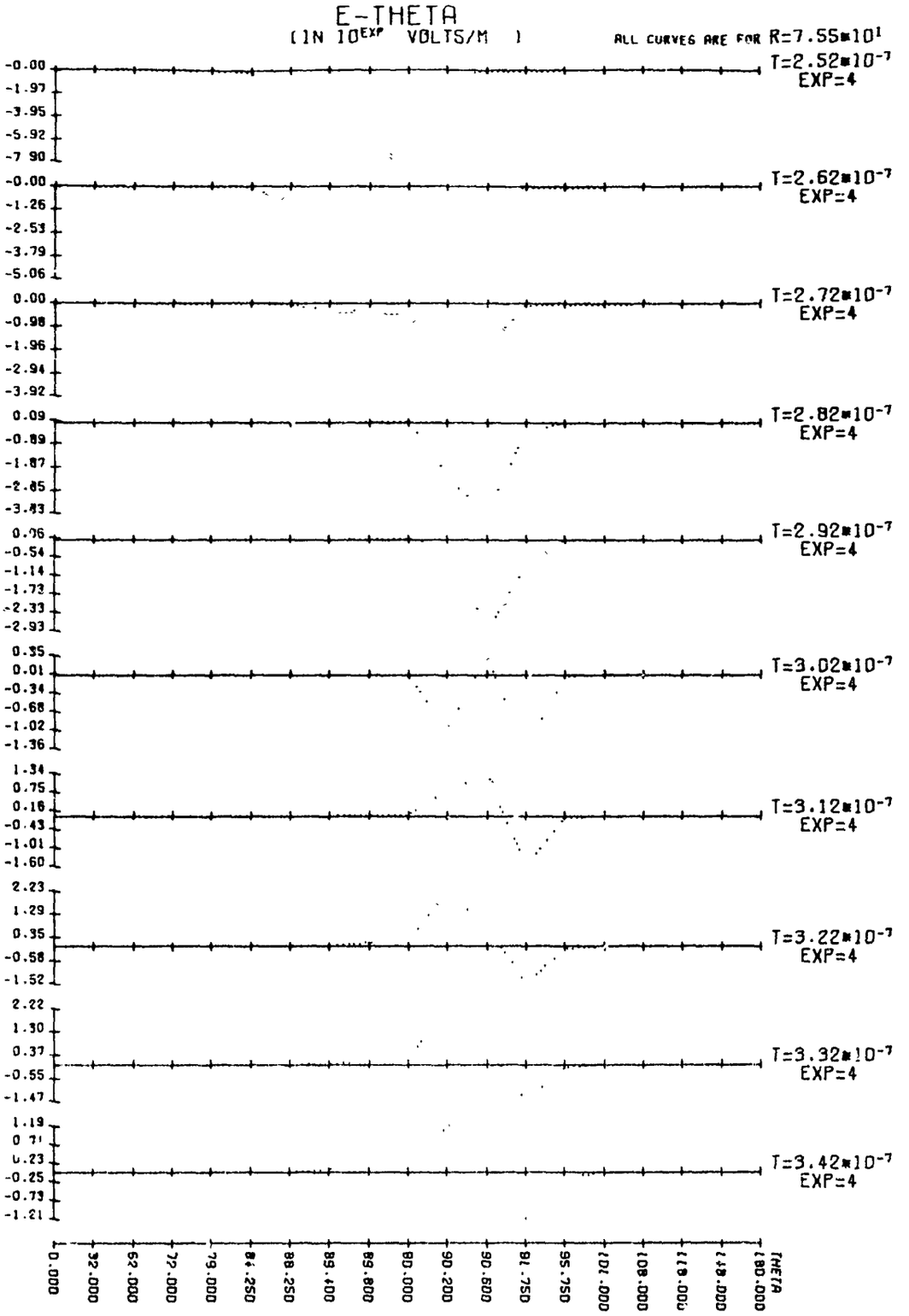

E-THETA
(IN 10^EXP VOLTS/M)
ALL CURVES ARE FOR R=7.55*10^1
T=2.52*10^-7
EXP=4
T=2.62*10^-7
EXP=4
T=2.72*10^-7
EXP=4
T=2.82*10^-7
EXP=4
T=2.92*10^-7
EXP=4
T=3.02*10^-7
EXP=4
T=3.12*10^-7
EXP=4
T=3.22*10^-7
EXP=4
T=3.32*10^-7
EXP=4
T=3.42*10^-7
EXP=4
THETA

Figure 4.

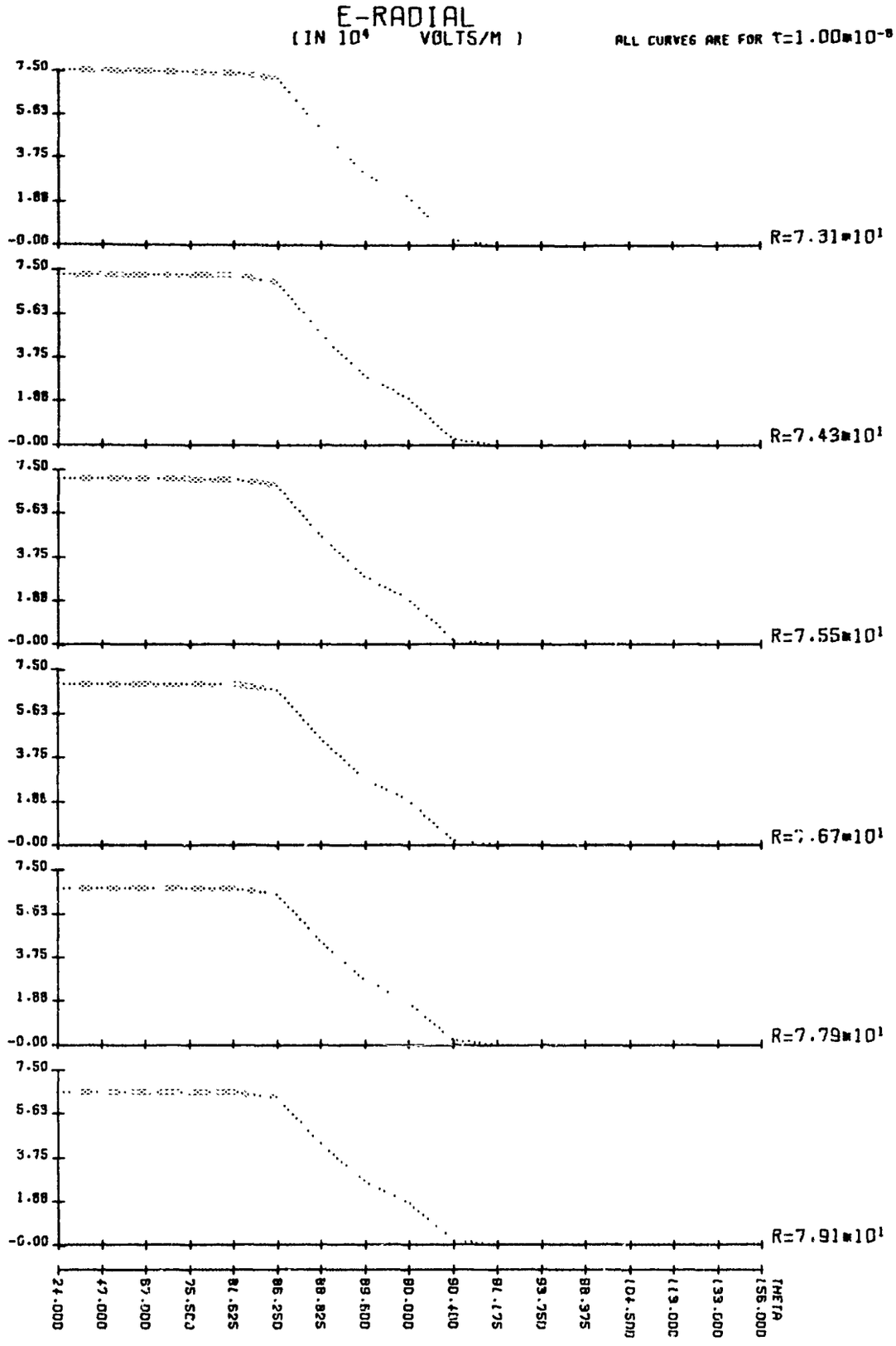

E-RADIAL
(IN 10⁴ VOLTS/M)
ALL CURVES ARE FOR τ=1.00*10⁻⁸
7.50
5.63
3.75
1.88
-0.00
R=7.31*10¹
R=7.43*10¹
R=7.55*10¹
R=7.67*10¹
R=7.79*10¹
R=7.91*10¹
24.000
42.000
57.000
75.563
84.525
90.250
88.825
89.600
90.000
90.400
91.175
93.750
98.375
104.500
119.000
133.000
156.000
THETA

Figure 5.

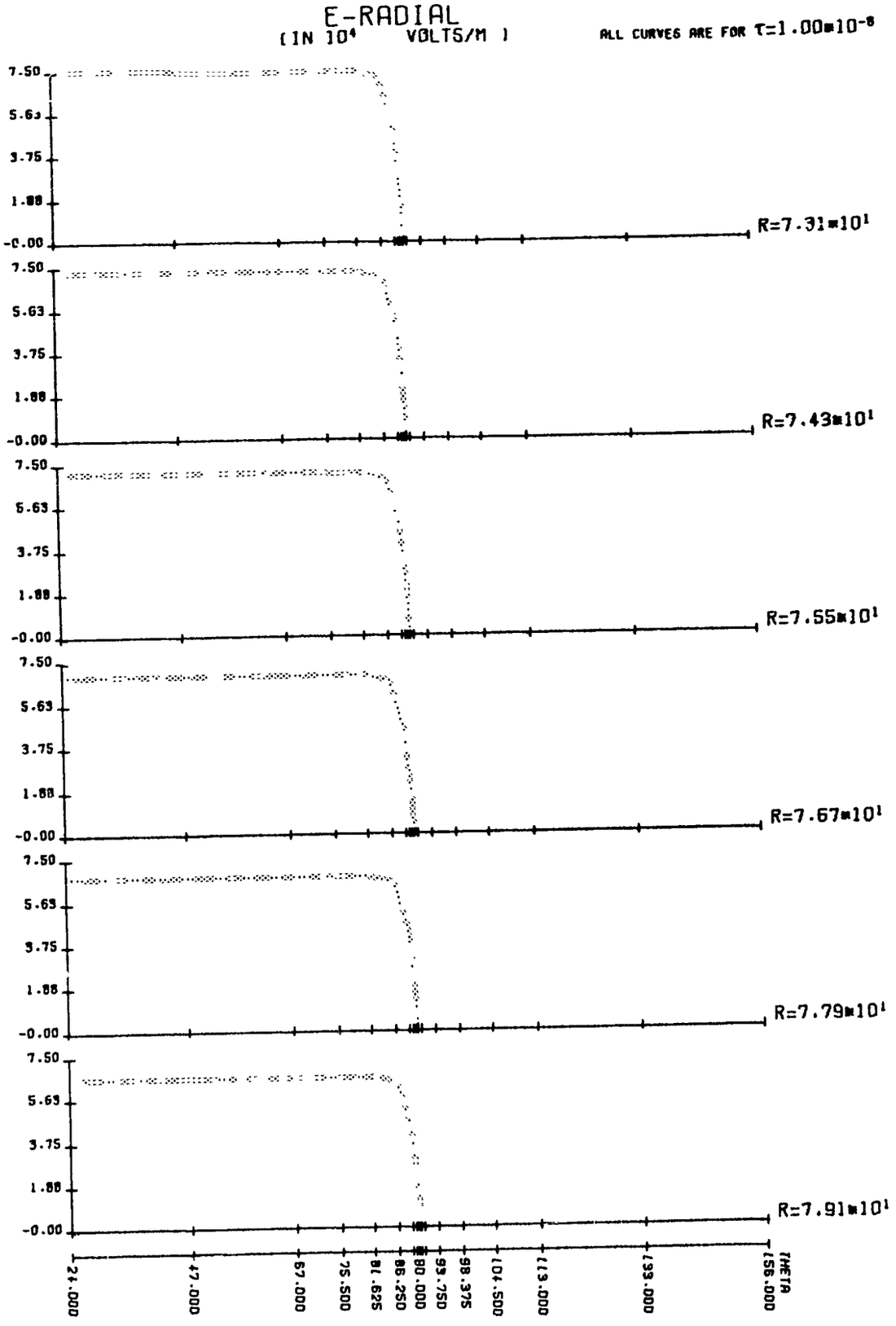

E-RADIAL
(IN 10⁴ VOLTS/M)
ALL CURVES ARE FOR τ=1.00*10⁻⁸
7.50
5.63
3.75
1.88
-0.00
R=7.31*10¹
R=7.43*10¹
R=7.55*10¹
R=7.67*10¹
R=7.79*10¹
R=7.91*10¹
21.000
47.000
67.000
75.500
81.625
86.250
90.000
93.750
98.375
104.500
113.000
133.000
159.000
THETA

Figure 6.

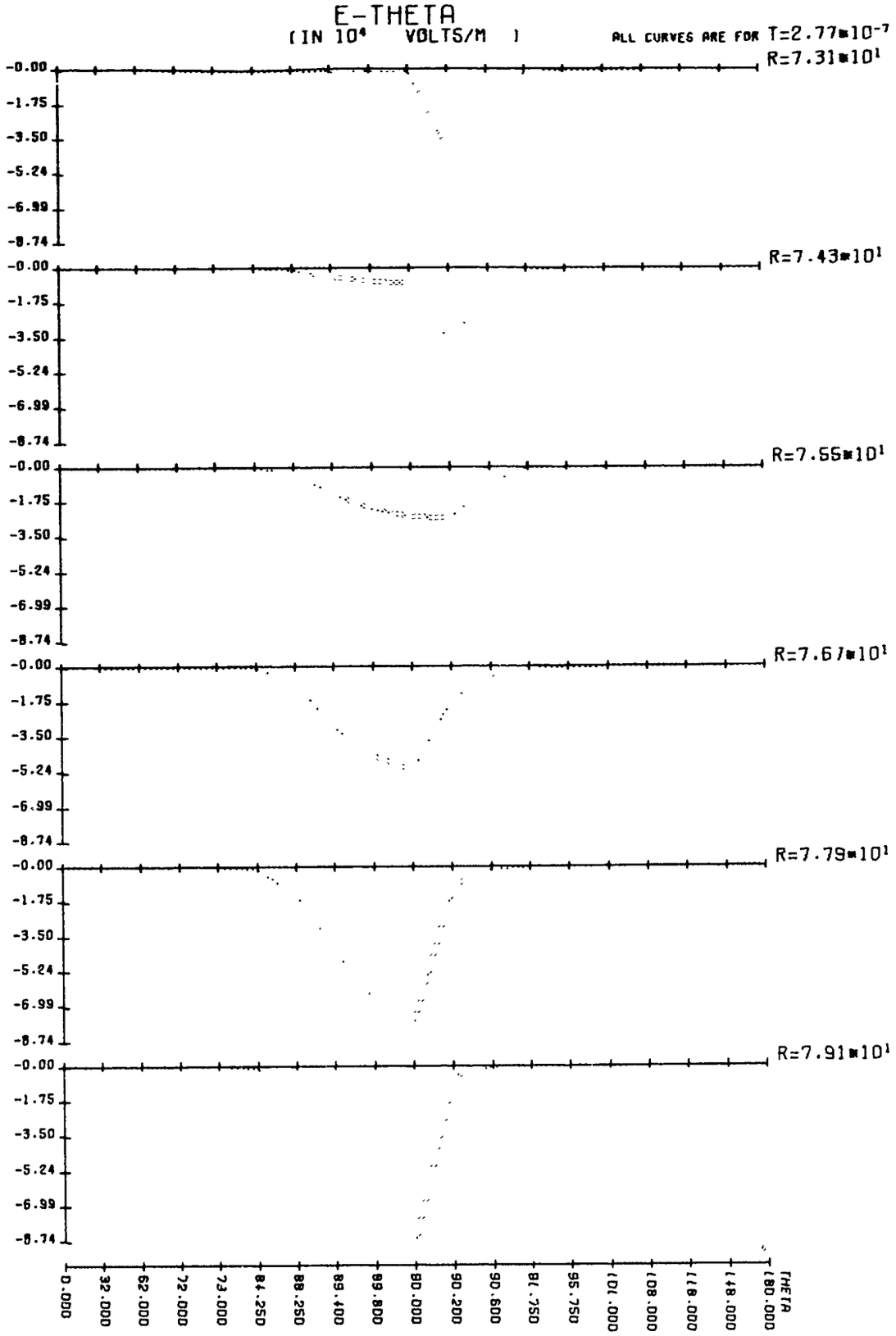

E-THETA
(IN 10⁴ VOLTS/M)
ALL CURVES ARE FOR T=2.77*10⁻⁷
R=7.31*10¹
R=7.43*10¹
R=7.55*10¹
R=7.67*10¹
R=7.79*10¹
R=7.91*10¹
-0.00
-1.75
-3.50
-5.24
-6.99
-8.74
0.000
32.000
62.000
72.000
73.000
84.250
88.250
89.400
89.800
90.000
90.200
90.600
91.750
95.750
101.000
108.000
118.000
180.000
THETA

Figure 7.

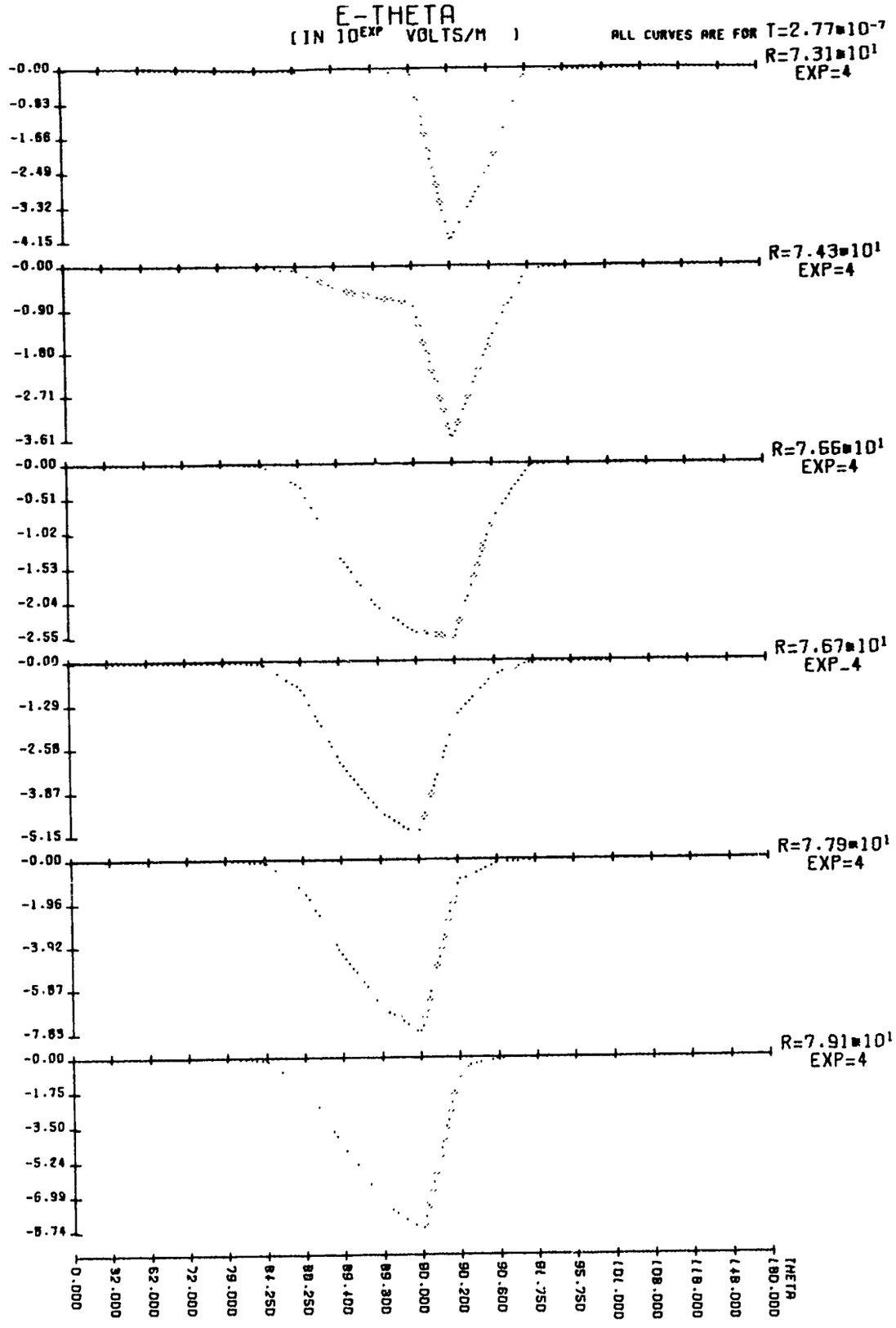

E-THETA
[IN 10^EXP VOLTS/M]
ALL CURVES ARE FOR T=2.77*10^-7
R=7.31*10^1
EXP=4
R=7.43*10^1
EXP=4
R=7.66*10^1
EXP=4
R=7.67*10^1
EXP=4
R=7.79*10^1
EXP=4
R=7.91*10^1
EXP=4
-0.00
-0.83
-1.66
-2.49
-3.32
-4.15
-0.00
-0.90
-1.80
-2.71
-3.61
-0.00
-0.51
-1.02
-1.53
-2.04
-2.55
-0.00
-1.29
-2.58
-3.87
-5.15
-0.00
-1.96
-3.92
-5.87
-7.83
-0.00
-1.75
-3.50
-5.24
-6.99
-8.74
0.000
32.000
62.000
72.000
78.000
84.250
88.250
89.400
89.800
90.000
90.200
90.600
91.750
95.750
101.000
108.000
118.000
148.000
180.000
THETA

Figure 8.

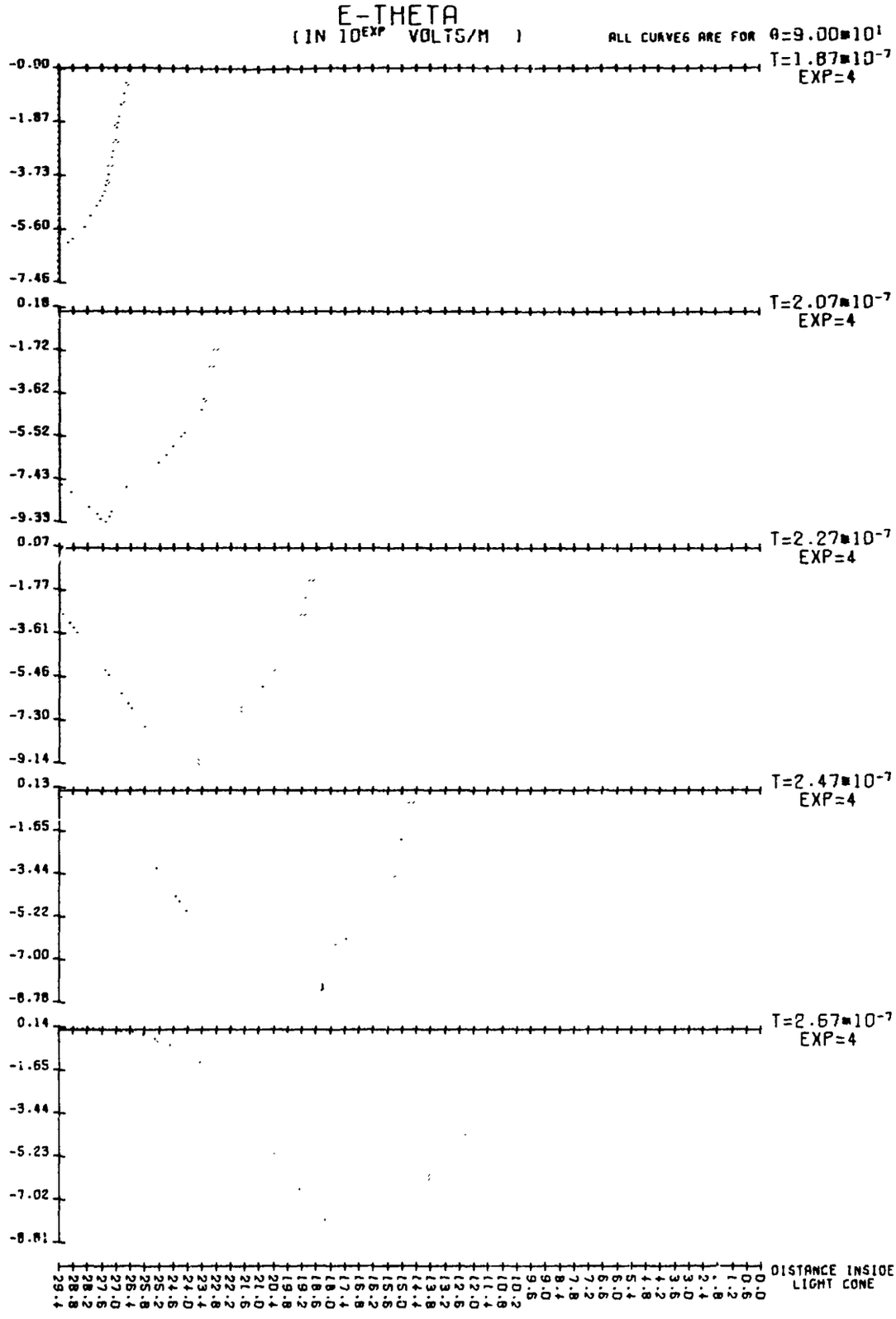

E-THETA
(IN 10^EXP VOLTS/M)
ALL CURVES ARE FOR θ=9.00*10^1
T=1.87*10^-7
EXP=4
-0.00
-1.87
-3.73
-5.60
-7.46
T=2.07*10^-7
EXP=4
0.18
-1.72
-3.62
-5.52
-7.43
-9.33
T=2.27*10^-7
EXP=4
0.07
-1.77
-3.61
-5.46
-7.30
-9.14
T=2.47*10^-7
EXP=4
0.13
-1.65
-3.44
-5.22
-7.00
-8.78
T=2.67*10^-7
EXP=4
0.14
-1.65
-3.44
-5.23
-7.02
-8.81
DISTANCE INSIDE
LIGHT CONE

Figure 9.

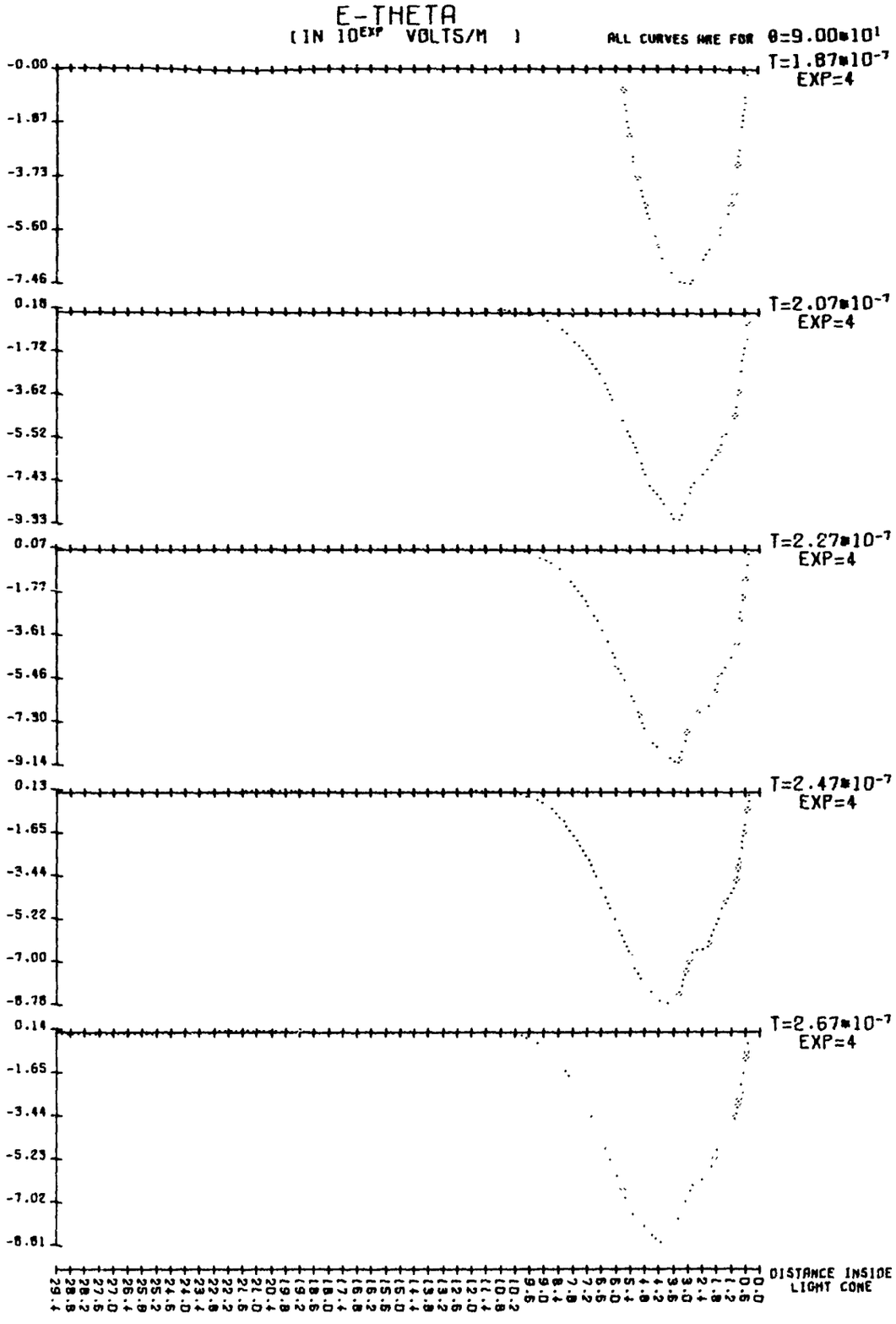

E-THETA
(IN 10^EXP VOLTS/M)
ALL CURVES ARE FOR θ=9.00*10^1
T=1.87*10^-7
EXP=4
T=2.07*10^-7
EXP=4
T=2.27*10^-7
EXP=4
T=2.47*10^-7
EXP=4
T=2.67*10^-7
EXP=4
DISTANCE INSIDE
LIGHT CONE

Figure 10.

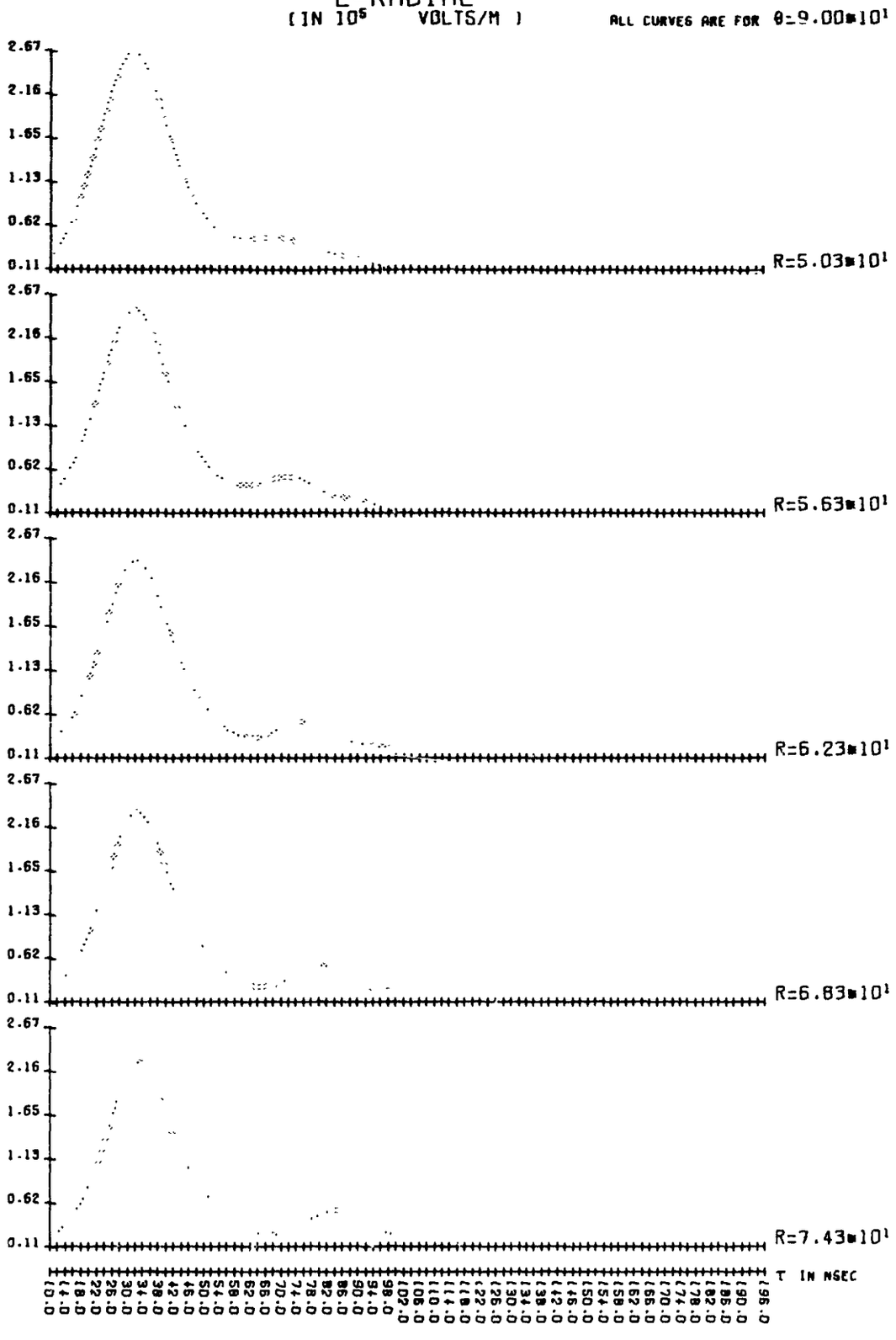

E-RADIAL
(IN 10⁵ VOLTS/M)
ALL CURVES ARE FOR θ=9.00*10¹
2.67
2.16
1.65
1.13
0.62
0.11
R=5.03*10¹
R=5.63*10¹
R=6.23*10¹
R=6.83*10¹
R=7.43*10¹
T IN NSEC

Figure 11.

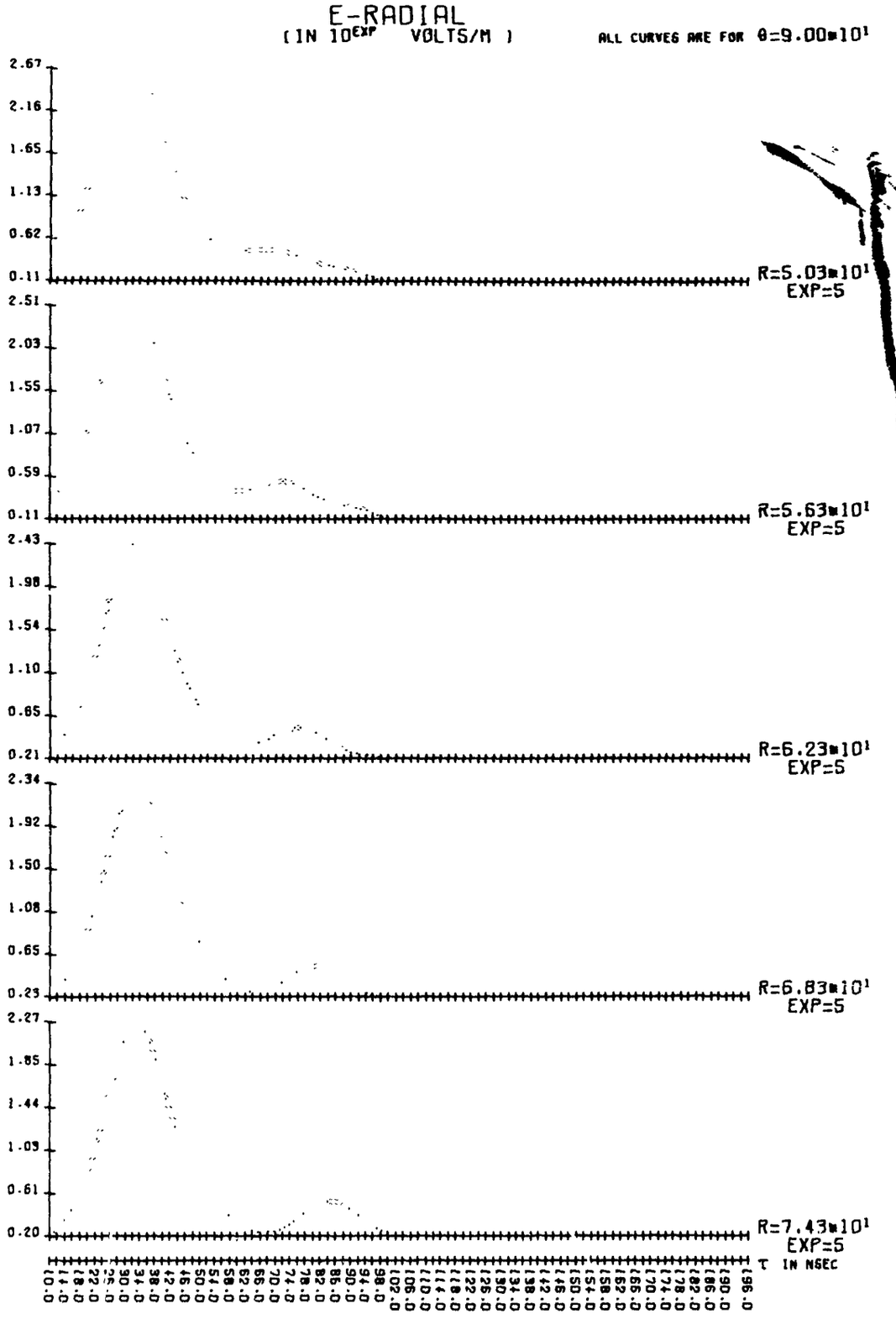

E-RADIAL
(IN 10EXP VOLTS/M)
ALL CURVES ARE FOR θ=9.00*10¹
R=5.03*10¹
EXP=5
R=5.63*10¹
EXP=5
R=6.23*10¹
EXP=5
R=6.83*10¹
EXP=5
R=7.43*10¹
EXP=5
τ IN NSEC

Figure 12.

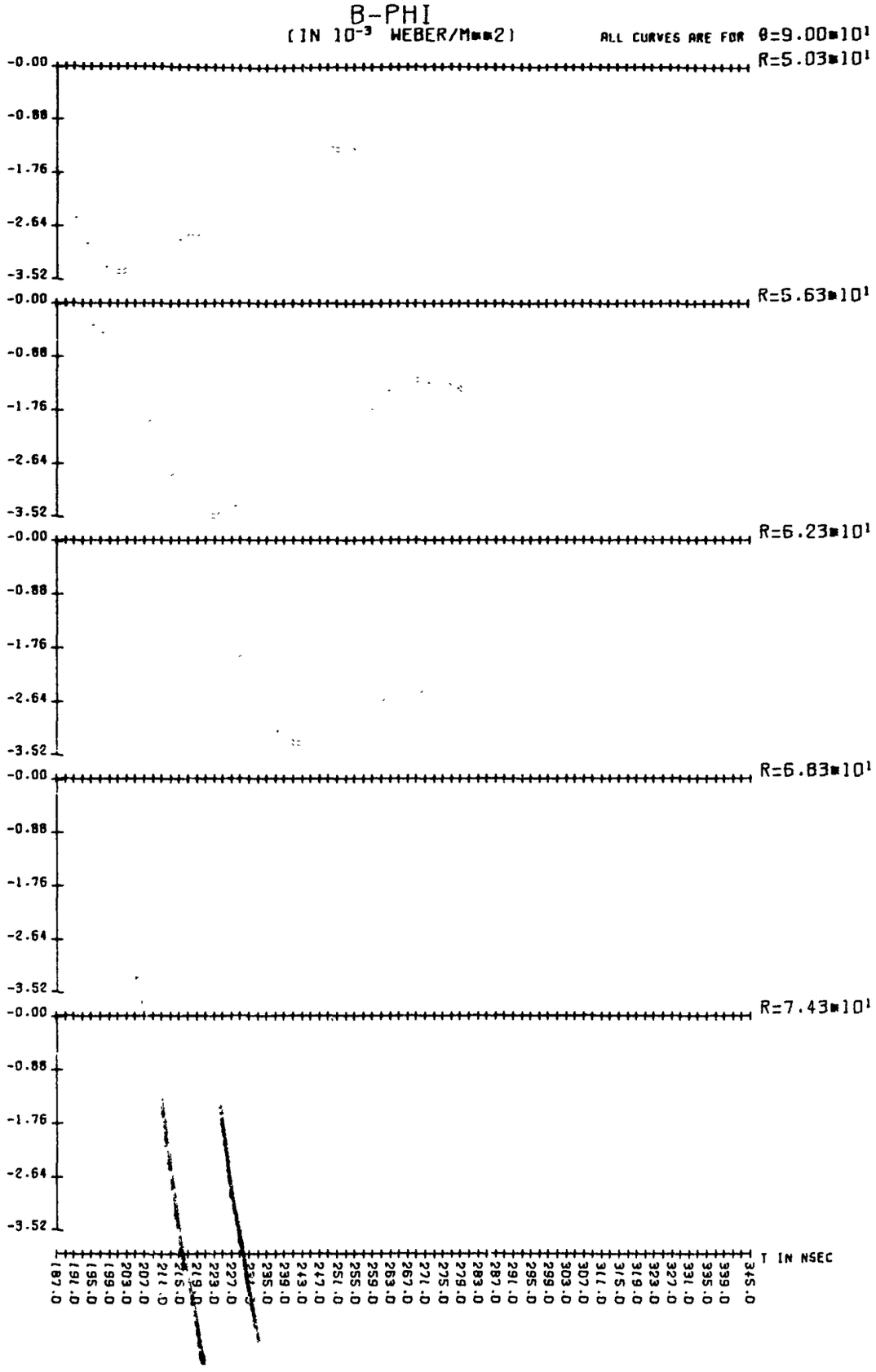

B-PHI
(IN 10⁻³ WEBER/M**2)
ALL CURVES ARE FOR θ=9.00*10¹
R=5.03*10¹
R=5.63*10¹
R=6.23*10¹
R=6.83*10¹
R=7.43*10¹
-0.00
-0.88
-1.76
-2.64
-3.52
T IN NSEC

Figure 13.

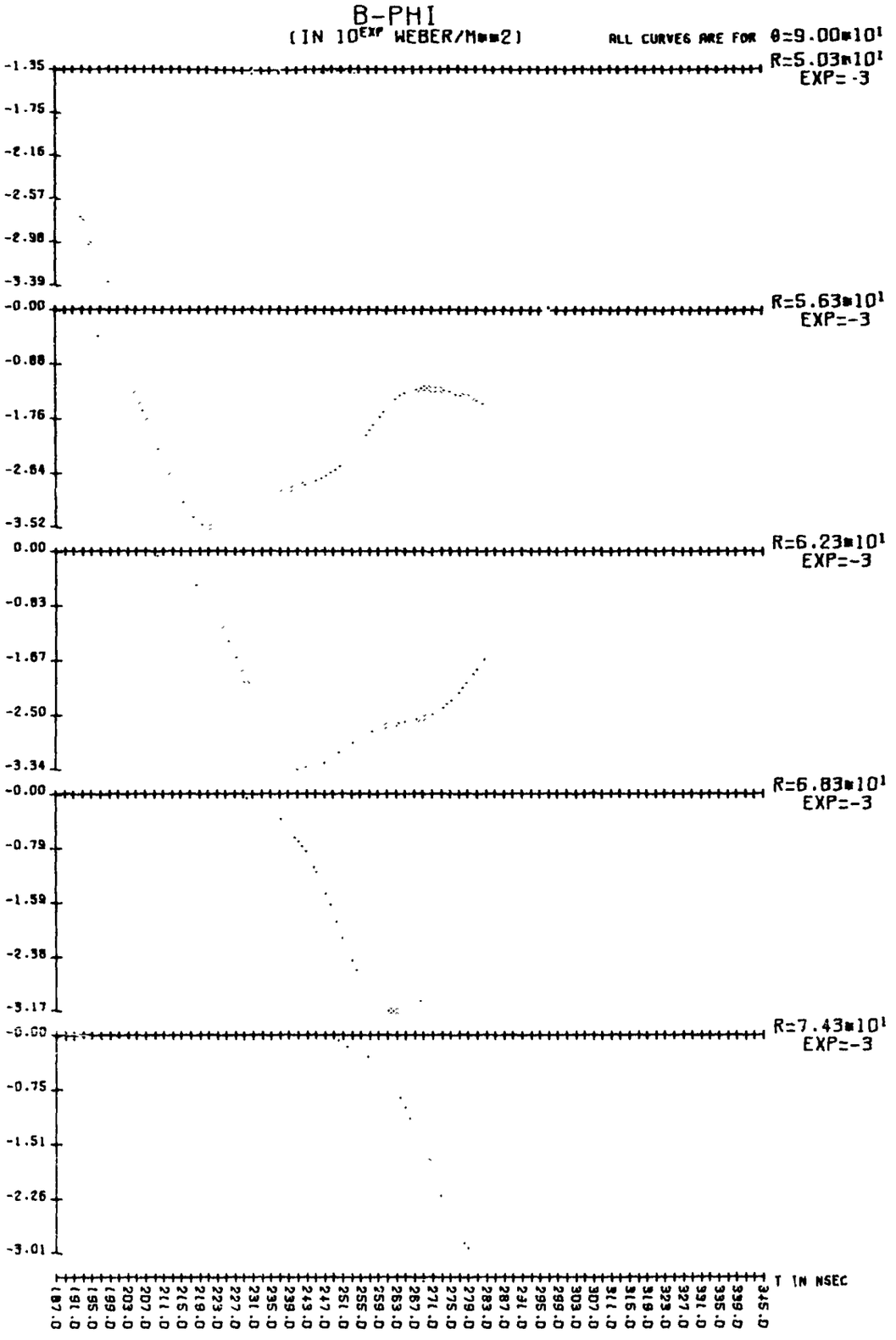

B-PHI
(IN 10^EXP WEBER/M**2)
ALL CURVES ARE FOR θ=9.00*10^1
R=5.03*10^1
EXP=-3
-1.35
-1.75
-2.16
-2.57
-2.98
-3.39
R=5.63*10^1
EXP=-3
-0.00
-0.88
-1.76
-2.64
-3.52
R=6.23*10^1
EXP=-3
0.00
-0.83
-1.67
-2.50
-3.34
R=6.83*10^1
EXP=-3
-0.00
-0.79
-1.59
-2.38
-3.17
R=7.43*10^1
EXP=-3
-0.00
-0.75
-1.51
-2.26
-3.01
T IN NSEC
187.0
191.0
195.0
199.0
203.0
207.0
211.0
215.0
219.0
223.0
227.0
231.0
235.0
239.0
243.0
247.0
251.0
255.0
259.0
263.0
267.0
271.0
275.0
279.0
283.0
287.0
291.0
295.0
299.0
303.0
307.0
311.0
315.0
319.0
323.0
327.0
331.0
335.0
339.0
343.0

Figure 14.

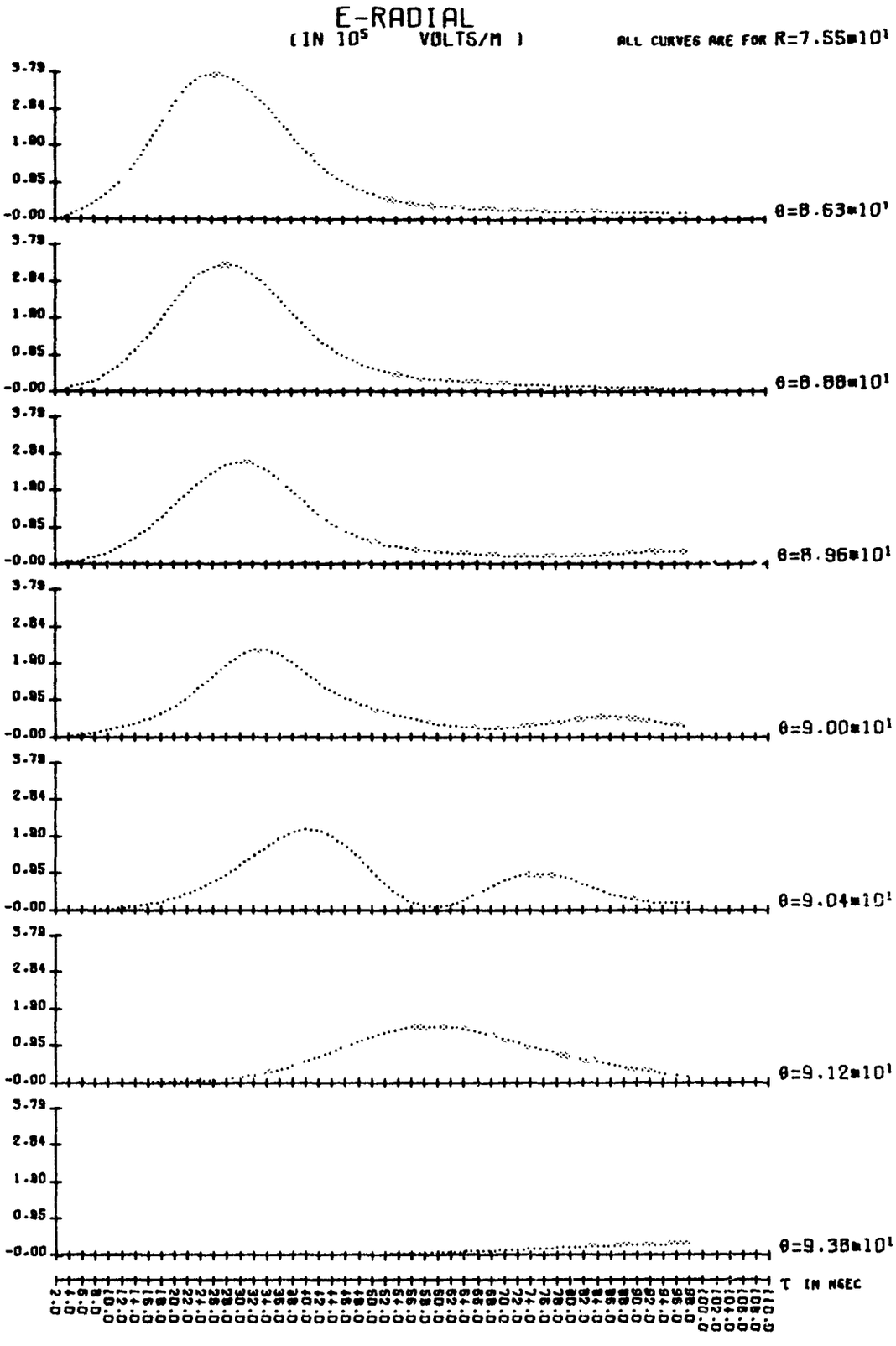

E-RADIAL
(IN 10^5 VOLTS/M)
ALL CURVES ARE FOR R=7.55*10^1
3.79
2.84
1.90
0.95
-0.00
θ=8.63*10^1
θ=8.88*10^1
θ=8.96*10^1
θ=9.00*10^1
θ=9.04*10^1
θ=9.12*10^1
θ=9.38*10^1
T IN NSEC

Figure 15.

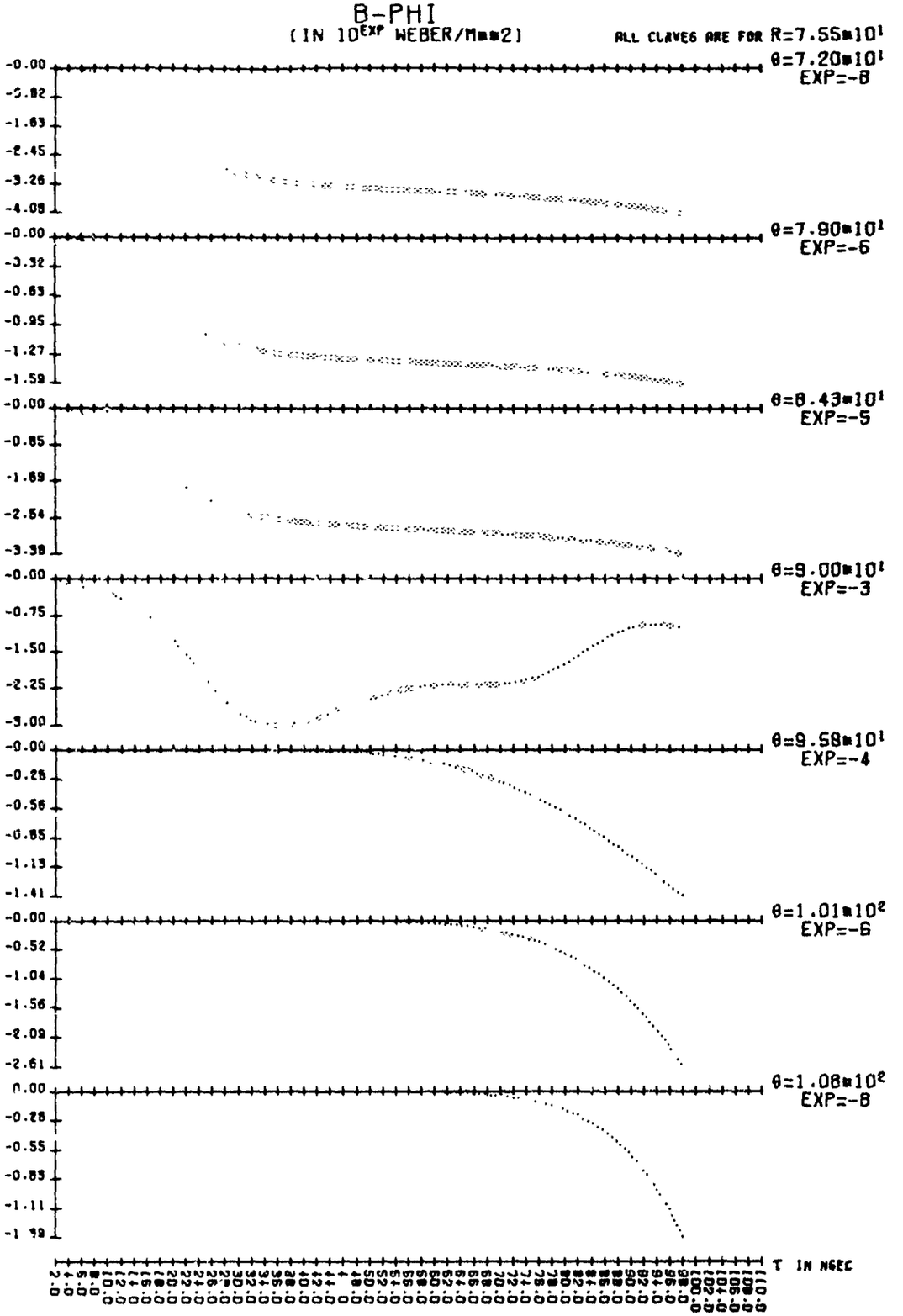

B-PHI
(IN 10^EXP WEBER/M**2)
ALL CURVES ARE FOR R=7.55*10^1
θ=7.20*10^1
EXP=-8
θ=7.80*10^1
EXP=-6
θ=8.43*10^1
EXP=-5
θ=9.00*10^1
EXP=-3
θ=9.58*10^1
EXP=-4
θ=1.01*10^2
EXP=-6
θ=1.08*10^2
EXP=-8
T IN NSEC

Figure 16.

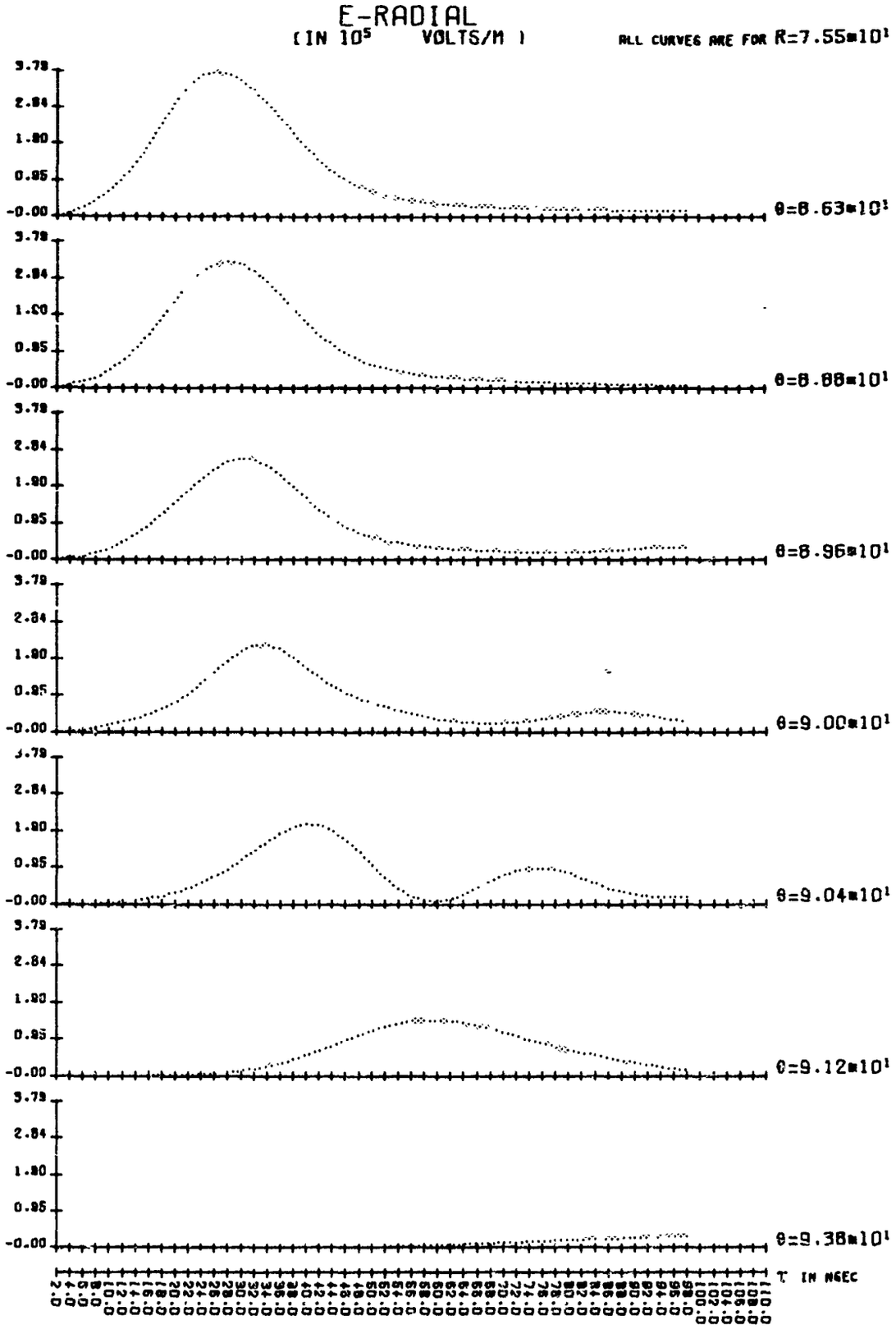

E-RADIAL
(IN 10⁵ VOLTS/M)
ALL CURVES ARE FOR R=7.55*10¹
3.78
2.84
1.90
0.95
-0.00
θ=8.63*10¹
θ=8.88*10¹
θ=8.96*10¹
θ=9.00*10¹
θ=9.04*10¹
θ=9.12*10¹
θ=9.38*10¹
τ IN MSEC

Figure 17.

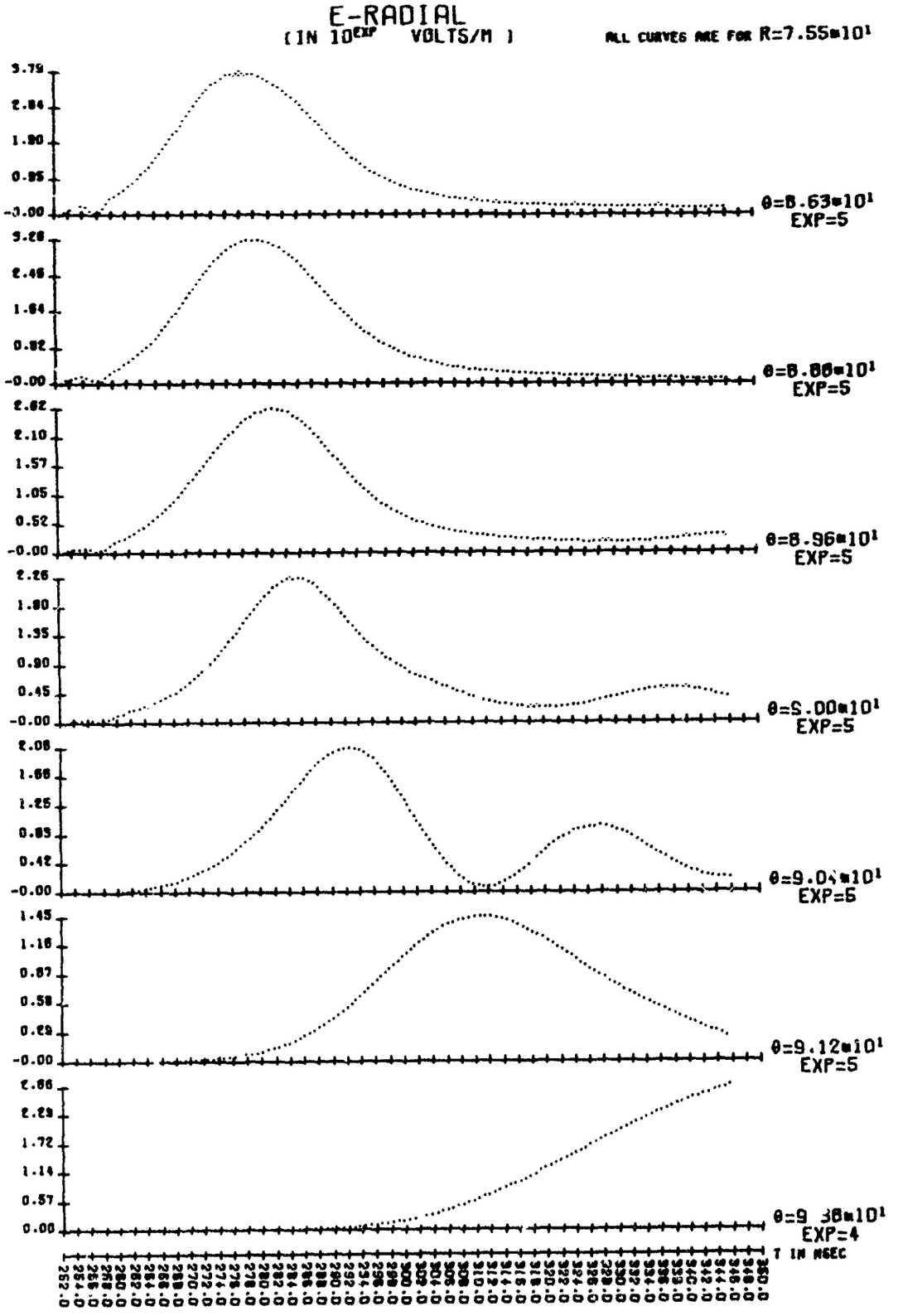

E-RADIAL
(IN 10^EXP VOLTS/M)
ALL CURVES ARE FOR R=7.55*10^1
θ=8.63*10^1
EXP=5
θ=8.80*10^1
EXP=5
θ=8.96*10^1
EXP=5
θ=9.00*10^1
EXP=5
θ=9.12*10^1
EXP=5
θ=9.38*10^1
EXP=4
T IN NSEC

Figure 18.

12. Program Listing

```
      PROGRAM COMPARE(INPUT,OUTPUT,TAPE1,TAPE2=4022,TAPE10)
000003 COMMON X(200),Y(2000),Z(10),XNUM(200),OUTARR(21,7),IBUF(1024),
     1 HEADER(8),NZ,NX,YLEN,XLEN,LAB,DT,LUN,LABTYP,XLAB,CONVAR,KONTYP,
     2 DELA,NPTS(10),TCOL
000003 DIMENSION TYPE(6),VAR(4),ICV(4),ITIME(4),LABEL(4),LUNITS(4),
     1 LBLTYP(5),XLABL(4),LOCTIME(10),IDATA(1)
000003 DATA TYPE/10HTIME-THETA,10HRANG-THETA,10HTIME-RANGE,10HRANGE-TIME,
     1 10HTHETA-TIME,10HSTOP       /, LABEL/10HSIGMA-TOT ,10H  R-PHI    ,
     2 10H E-THETA  ,10H E-RADIAL /, LUNITS/10H  MHO/M   ,10HWEBER/M**2,
     3 10H VOLTS/M  ,10H  VOLTS/M /, XLABL/5HTHETA,5HTHETA,5HRANGE,
     4 5HTIME /, VAR/4HSIGT,4HPHI,4HETHE,4HERAD/, LBLTYP/2HT=,2HR=,2HT=
     5,2HR=,2H  /, ICV/3,4,5,7/, ITIME/3,4,5,7/
000003  1 FORMAT(A10)
000003  2 FORMAT(I10,7F10.0)
000003  3 FORMAT(A4,6X,2F10.0,2I10)
000003  4 FORMAT(8F10.0)
000003  5 FORMAT(A4,6X,F10.0,2I10,F10.0)
000003  9 FORMAT(* THETA VALUE*,F10.3,*NOT VALID FOR THIS DATA TAPE*)
000003 10 FORMAT(/* CHOICE *,A10,* DOES NOT MATCH ANY MEMBER OF TYPE ARRAY*)
000003    CALL PLOTS(IBUF,1024,10)
000006    CALL PLOT(0.0,-11.,-3)
000011    IDATA(1)=1
000012    CALL FTNBIN(1,1,IDATA)
000015 100 READ 1, CHOICE
000023    DO 105 ITYPE=1,6
000025    IF(CHOICE.EQ.TYPE(ITYPE)) GO TO 110
000027 105 CONTINUE
000031    PRINT 10, CHOICE
000037    GO TO 100
000040 110 GO TO (200,200,1000,2000,3000,4000), ITYPE

          C
          C       X-AXIS = THETA       Z-AXIS = TIME        CONSTANT = RANGE
          C                                   OR
          C       X-AXIS = THETA       Z-AXIS = RANGE       CONSTANT = TIME
          C

000052 200 READ 2, NZ
000060    READ 4, (Z(I),I=1,NZ)
000073    READ 4, DT,DR,CONVAR,TCOL,XLEN,YLEN
000113    REWIND 1
000115    IUNIT=1
000116    IWROTE=0
000117    XLAB=XLABL(ITYPE)
000121    LABTYP=LBLTYP(ITYPE)
000123    KONTYP=LBLTYP(ITYPE+1)
000124    HDR=.5*DR
000126 220 READ 3, AVAR,THETMN,THETMX,INCX,ISCALE
000144    IF(INCX.EQ.0) GO TO 100
000145    REWIND 2
000147    DO 23  J=1,4
000151    IC=J
000152    IF(AVAR.EQ.VAR(J)) GO TO 235
000154 230 CONTINUE
000156 235 READ(IUNIT) NDUM,NDUM,LMAX
000167    READ(IUNIT) HEADER
000174    READ(IUNIT) OUTARR
000201    IF(IWROTE.EQ.1) GO TO 240
000203    WRITE(2) NDUM,NDUM,LMAX
```

```
000214          WRITE(2) HEADER
000221          WRITE(2) OUTARR
000226      240 LAB=LABEL(IC)
000230          LUN=LUNITS(IC)
000232          ITHETA=0
000233          ICOL=1
000234          LIMIT=LMAX+1
000236          IF(IC.NE.4) GO TO 245
000240          ICOL=6
000241          LIMIT=LMAX
000242      245 DO 250 I=1,LIMIT
000244          IP3=I+3
000246          TEMP=OUTARR(IP3,ICOL)
000252          IF(TEMP.LT.THETMN) GO TO 250
000254          IF(TEMP.GT.THETMX) GO TO 250
000260          ITHETA=ITHETA+1
000261          IF(ITHETA.EQ.1) ITMIN=I
000264          XNUM(ITHETA)=TEMP
000266      250 CONTINUE
000271          I90=0
000272          IF(IC.EQ.4) GO TO 280
000274          DO 260 I=1,ITHETA
000275          IF(XNUM(I).GE.90.) GO TO 265
000300      260 CONTINUE
000302          GO TO 280
000303      265 I90=I
000305          I90M1=I90-1
000307          I90P1=I90+1
000310          IPUT=ITHETA
000311          DO 270 J=I90,ITHETA
000312          XNUM(IPUT+1)=XNUM(IPUT)
000314          IPUT=IPUT-1
000316      270 CONTINUE
000320          XNUM(I90)=90.
000322          ITHETA=ITHETA+1
000323      280 TL=0
000324          INDEX=0
000325          NX=ITHETA
000326          NXZ=NX*N7
000330          DO 290 K=1,NXZ
000332          Y(K)=0.0
000333      290 CONTINUE
000335          LSTART=1
000336          ITIM=ITIME(IC)
000340          IF(ICOL.NE.0.) ITIM=1
000342          ICVM=ICV(IC)
000344          IF(ITYPE.EQ.2) GO TO 360
000346          GO TO 320

C         X-AXIS = THETA       Z-AXIS = TIME        CONSTANT = RANGE

000346      300 READ(IUNIT) OUTARR
000353          IF(EOF,IUNIT) 500,320
000356      320 IF(OUTARR(1,6).LT.CONVAR-HDR) GO TO 300
000362          IF(OUTARR(1,6).GT.CONVAR+HDR) GO TO 300
000366          T=OUTARR(1,ITIM)
000371          DO 330 L=LSTART,NZ
000373          IF(T-DT.GT.Z(L)) GO TO 330
```

```
000400          IF(T+DT.GE.Z(L)) GO TO 335
000402          GO TO 330
000403      330 CONTINUE
000406          GO TO 300
000406      335 IL=IL+1
000410          LSTART=L
000411          IF(IWROTF.EQ.2) WRITE(2) OUTARR
000417          IF(I90.EQ.0) GO TO 342
000420          DO 338 M=1,I90M1
000422          INDEX=INDEX+1
000424          IR=ITMIN+2+M
000426          Y(INDEX)=OUTARR(IR,ICVM)
000433      338 CONTINUE
000435          INDEX=INDEX+1
000437          Y(INDEX)=OUTARR(2,ICV4)
000443          DO 340 MM=I90P1,ITHETA
000445          INDEX=INDEX+1
000447          IR=ITMIN+1+MM
000451          Y(INDEX)=OUTARR(IR,ICVM)
000456      340 CONTINUE
000460          IF(IL.LT.NZ) GO TO 300
000463          GO TO 500
000463      342 DO 345 M=1,ITHETA
000465          INDEX=INDEX+1
000467          IR=ITMIN+2+M
000471          Y(INDEX)=OUTARR(IR,ICVM)
000476      345 CONTINUE
000500          IF(IL.LT.NZ) GO TO 300
000503          GO TO 500

                   X-AXIS = THETA      Z-AXIS = RANGE      CONSTANT = TIME

000503      350 READ(IUNIT) OUTARR
000510          IF(EOF,IUNIT) 500,360
000513      360 IF(OUTARR(1,ITIM)-DT.GT.CONVAR) GO TO 350
000522          IF(OUTARR(1,ITIM)+DT.LT.CONVAR) GO TO 350
000526          DO 370 L=LSTART,NZ
000530          IF(OUTARR(1,6).LT.Z(L)-HDR) GO TO 350
000534          IF(OUTARR(1,6).LE.Z(L)+HDR) GO TO 380
000537      370 CONTINUE
000542          GO TO 350
000542      380 IL=IL+1
000544          LSTART=L
000545          IF(IWROTF.EQ.2) WRITE(2) OUTARR
000553          IF(I90.EQ.0) GO TO 410
000554          DO 390 M=1,I90M1
000556          INDEX=INDEX+1
000560          IR=ITMIN+2+M
000562          Y(INDEX)=OUTARR(IR,ICVM)
000567      390 CONTINUE
000571          INDEX=INDEX+1
000573          Y(INDEX)=OUTARR(2,ICVM)
000577          DO 400 MM=I90P1,ITHETA
000601          INDEX=INDEX+1
000603          IR=ITMIN+1+MM
000605          Y(INDEX)=OUTARR(IR,ICVM)
000612      400 CONTINUE
000614          IF(IL.LT.NZ) GO TO 35
```

```
000617          GO TO 500
000617      410 DO 450 M=1,ITHETA
000621          INDEX=INDEX+1
000623          IR=ITMIN+2+M
000625          Y(INDEX)=OUTARR(IR,ICVM)
000632      450 CONTINUE
000634          IF(IL.LT.NZ) GO TO 350
000637      500 IF(INCX.EQ.2) GO TO 600
000641          DO 550 I=1,ITHETA
000643          X(I)=XNUM(I)
000645      550 CONTINUE
000647          GO TO 700
000647      600 DO 650 I=1,ITHETA
000651          X(I)=I
000653      650 CONTINUE
000655      700 IF(ISCALF.EQ.0) GO TO 750
000656          CALL LYNEUP
000657          GO TO 800
000660      750 CALL LINEUP
000661      800 CALL PLOT(XLEN+2.5,0.0,-3)
000666          IUNIT=2
000667          IWRITF=1
000670          GO TO 220

C               X-AXIS = RET.RANGE          Z-AXIS = TIME        CONSTANT = THETA

000671     1000 READ 2, NZ
000677          READ 4, (Z(I),I=1,NZ)
000712          TMAX=Z(1)
000714          DO 1005 I=2,NZ
000715          IF(TMAX.LT.Z(I)) TMAX=Z(I)
000721     1005 CONTINUE
000724          READ 4, DT,DR,XLEN,YLEN
000737          HDR=.5*DR
000741          READ 2, NRANGE,RMIN,RINC
000753          XNUM(1)=RMIN
000755          DO 1010 M=2,NRANGE
000756          XNUM(M)=XNUM(M-1)+RINC
000761     1010 CONTINUE
000763          DELX=(NRANGE-1)*RINC
000767          NPZ=NRANGE*NZ
000771          NX=NRANGE
000772          XLAB=XLABL(ITYPE)
000774          LABTYP=LBLTYP(ITYPE)
000776     1030 READ 5, AVAR,CONVAR,INCX,ISCALE
001012          IF(INCX.EQ.0) GO TO 100
001013          DO 1035 I=1,NPZ
001015          Y(I)=0.0
001016     1035 CONTINUE
001020          DO 1040 J=1,4
001022          IC=J
001023          IF(AVAR.EQ.VAR(J)) GO TO 1050
001025     1040 CONTINUE
001027     1050 RE=IND 1
001031          READ(1) NDUM,NDUM,LMAX
001042          READ(1) HEADER
001047          READ(1) OUTARR
001054          ICVM=ICV(IC)
```

```
001056          LAB=LABEL(IC)
001040          LUN=LUNITS(IC)
001041          INDEX=0
001042          TLAST=0.
001043          IR=0
001044          ICOL=6
001045          LIMIT=LMAX
001066          IF(IC.EQ.4) GO TO 1055
001070          ICOL=1
001071          LIMIT=LMAX+1
001073          IF(CONVAR.EQ.90.) IR=2
001076    1055  IF(IR.EQ.2) GO TO 1070
001100          DO 1060 I=1,LIMIT
001102          IR=I+3
001104          IF(OUTARR(IR,ICOL).EQ.CONVAR) GO TO 1070
001111    1060  CONTINUE
001113          PRINT 9, CONVAR
001121          GO TO 1030
001122    1065  READ(1) OUTARR
001127          IF(EOF,1) 1200,1070
001132    1070  T=OUTARR(1,1)
001134          IF(T.EQ.TLAST) GO TO 1100
001136          DO 1080 I=1,N7
001137          IF(T-DT.GT.Z(I)) GO TO 1080
001144          IF(T+DT.GE.Z(I)) GO TO 1100
001146    1080  CONTINUE
001151          IF(T-DT.GT.TMAX) GO TO 1200
001155          GO TO 1065
001155    1100  R=OUTARR(1,6)
001167          TLAST=T
001160          DO 1110 M=1,NRANGE
001162          IF(R-HDR.GT.XNUM(M)) GO TO 1110
001167          IF(R+HDR.GE.XNUM(M)) GO TO 1130
001171    1110  CONTINUE
001174          GO TO 1065
001174    1130  INDEX=(L-1)*NRANGE+M
001201          Y(INDEX)=OUTARR(IR,ICVM)
001206          GO TO 1065
001206    1200  IF(INCX.EQ.2) GO TO 1220
001210          DO 1210 I=1,NRANGE
001212          X(I)=XNUM(I)
001214    1210  CONTINUE
001216          GO TO 1240
001216    1220  DO 1230 I=1,NRANGE
001220          X(I)=T
001222    1230  CONTINUE
001224    1240  IF(ISCALF.EQ.0) GO TO 1250
001225          CALL LTKONE
001226          GO TO 1300
001227    1250  CALL LTCONE
001230    1300  CALL PLOT(XLEN+2.5,0.,-3)
001235          GO TO 1030

C         X-AXIS = TIME       Z-AXIS = RANGE      CONSTANT = THETA

001236    2000  READ 2, NZ
001244          READ 4, (Z(I),I=1,NZ)
001257          READ 4, DT,DR,XLEN,YLEN
```

```
001273          HDR=.5*DR
001275          READ 2, NDELTS,TMIN
001305          NX=1
001306          XNUM(1)=TMIN
001310          DO 2010 I=1,NDELTS
001311          READ 2, NDELT,DELT
001320          DO 2010 J=1,NDELT
001322          NX=NX+1
001324          XNUM(NX)=XNUM(NX-1)+DELT
001327    2010 CONTINUE
001334          REWIND 1
001336          IUNIT=1
001337          IWROTE=0
001340          NXZ=NX*N7
001343          XLAB=XLABL(ITYPE)
001345          LABTYP=LBLTYP(ITYPE)
001346          KONTYP=LBLTYP(ITYPE+1)
001350    2030 READ 5, AVAR,CONVAR,INCX,ISCALE,TCOL
001366          IF(INCX.EQ.0) GO TO 100
001367          REWIND 2
001371          DO 2032 I=1,N7
001373          NPTS(I)=0
001374    2032 CONTINUE
001375          DO 2 35 I=1,NXZ
001400          Y(I)=0.0
001401    2035 CONTINUE
001403          DO 2 40 J=1,4
001405          IC=J
001406          IF(AVAR.EQ.VAR(J)) GO TO 2050
001410    2040 CONTINUE
001412    2050 READ(IUNIT) NDUM,NDUM,LMAX
001423          READ(IUNIT) HEADER
001430          READ(IUNIT) OUTARR
001435          IF(IWROTE.EQ.1) GO TO 2055
001437          WRITE(2) NDUM,NDUM,LMAX
001440          WRITE(2) HEADER
001455          WRITE(2) OUTARR
001442    2055 ICVM=ICV(IC)
001444          LAB=LABEL(IC)
001446          LUN=LUNITS(IC)
001447          ITIM=ITIME(IC)
001471          IF(TCOL.NE.0.) ITIM=1
001473          IL=0
001474          IR=0
001475          ICOL=6
001476          LIMIT=LMAX
001477          IF(IC.EQ.4) GO TO 2060
001501          ICOL=1
001502          LIMIT=LMAX+1
001504          IF(CONVAR.EQ.90.) IR=2
001507    2060 IF(IR.EQ.2) GO TO 2075
001511          DO 2065 I=1,LIMIT
001513          IR=I+3
001515          IF(OUTARR(IR,ICOL).EQ.CONVAR) GO TO 2075
001522    2065 CONTINUE
001524          PRINT 9, CONVAR
001532          GO TO 2030
001533    2070 READ(IUNIT) OUTARR
```

```
001540          IF(EOF.IUNIT) 2120,2075
001543     2075 R=OUTARR(1,6)
001545          DO 2080 M=1,NZ
001546          IF(R-HDR.GT.Z(M)) GO TO 2080
001553          IF(R+HDR.GE.Z(M)) GO TO 2090
001555          GO TO 2070
001556     2080 CONTINUE
001561          GO TO 2070
001561     2090 T=OUTARR(1,ITIM)
001565          DO 2100 L=1,NX
001566          IF(T-DT.GT.XNUM(L)) GO TO 2100
001573          IF(T+DT.GE.XNUM(L)) GO TO 2110
001575          GO TO 2070
001576     2100 CONTINUE
001601          IF(M.EQ.NZ.AND.L.EQ.NX) GO TO 2120
001610          GO TO 2070
001611     2110 INDEX=(M-1)*NX+L
001616 6        Y(INDEX)=OUTARR(IR,ICVM)
001623          IF(IWROTE.EQ.0) WRITE(2) OUTARR
001631          NPTS(M)=NPTS(M)+1
001634          IL=IL+1
001635          IF(IL.LT.NXZ) GO TO 2070
001637     2120 IF(INCX.EQ.2) GO TO 2135
001641          DO 2130 I=1,NX
001643          X(I)=XNUM(I)
001645     2130 CONTINUE
001647          GO TO 2150
001647     2135 DO 2140 I=1,NX
001651          X(I)=I
001653     2140 CONTINUE
001655     2150 IF(ISCALF.EQ.0) GO TO 2160
001656          CALL TYMAXS
001657          GO TO 2200
001660     2160 CALL TIMAXS
001661     2200 CALL PLOT(XLEN+2.5,0.0,-3)
001666          IUNIT=2
001667          IWROTE=1
001670          GO TO 2030

C         X-AXIS = TIME        Z-AXIS = THETA       CONSTANT = RANGE

001671     3000 READ 4, DT,DR,XLEN,YLEN,CONVAR
001707          HDR=.5*DR
001711          READ 2, NDELTS,TMIN
001721          NX=1
001722          XNUM(1)=TMIN
001724          DO 3010 I=1,NDELTS
001725          READ 2, NDELT,DELT
001734          DO 3010 J=1,NDELT
001736          NX=NX+1
001740          XNUM(NX)=XNUM(NX-1)+DELT
001743     3010 CONTINUE
001750          REWIND 1
001752          IUNIT=1
001753          IWROTE=0
001754          KONTYP=LBLTYP(2)
001756          LABTYP=LBLTYP(ITYPE)
001760     3020 READ 2, NZ
```

```
001766          READ 4, (Z(I),I=1,N7)
002001    3030  READ 5, AVAR,TCOL,INCX,ISCALE
002015          IF(INCX.EQ.0) GO TO 100
002016          IF(INCX.EQ.3) GO TO 3020
002020          REWIND 2
002022          DO 3032 I=1,N7
002024          NPTS(I)=0
002025    3032  CONTINUE
002027          NXZ=NX*N7
002031          DO 3035 I=1,NXZ
002033          Y(I)=0.0
002034    3035  CONTINUE
002036          DO 3040 J=1,4
002040          IC=J
002041          IF(AVAR.EQ.VAR(J)) GO TO 3050
002043    3040  CONTINUE
002045    3050  READ(IUNIT) NDUM,NDUM,LMAX
002056          READ(IUNIT) HEADER
002063          READ(IUNIT) OUTARR
002070          IF(IWROTE.EQ.1) GO TO 3055
002072          WRITE(2) NDUM,NDUM,LMAX
002103          WRITE(2) HEADER
002110          WRITE(2) OUTARR
002115    3055  ICVM=ICV(IC)
002117          LAB=LABEL(IC)
002121          LUN=LUNITS(IC)
002122          ITIM=ITIME(IC)
002124          IF(TCOL.NE.0.) ITIM=1
002126          I6=0
002127          ICOL=6
002130          LIMIT=LMAX
002132          IF(IC.EQ.4) GO TO 3057
002134          ICOL=1
002135          LIMIT=LMAX+1
002137    3057  DO 3065 J=1,N7
002141          ZTEST=Z(J)
002143          IF(ZTEST.NE.99.) GO TO 3058
002145          IF(IC.EQ.4) GO TO 3058
002147          LOCTHE(J)=2
002151          GO TO 3065
002151    3058  DO 3060 I=1,LIMIT
002153          IP3=I+3
002155          TEMP=OUTARR(IP3,ICOL)
002161          IF(TEMP.NE.ZTEST) GO TO 3060
002163          LOCTHE(J)=IP3
002165          GO TO 3065
002165    3060  CONTINUE
002170    3065  CONTINUE
002173          GO TO 3075
002173    3070  READ(IUNIT) OUTARR
002200          IF(EOF,IUNIT) 3200,3075
002203    3075  R=OUTARR(1,6)
002205          IF(R-HDR.GT.CONVAR) GO TO 3070
002211          IF(R+HDR.LT.CONVAR) GO TO 3070
002213          T=OUTARR(1,ITIM)
002217          DO 3080 L=1,NX
002220          IF(T-DT.GT.XNUM(L)) GO TO 3080
002225          IF(T+DT.GE.XNUM(L)) GO TO 3090
```

```
002227          GO TO 3070
002230    3080  CONTINUE
002233          GO TO 3200
002233    3090  IF(IWROTF.EQ.0) WRITE(2) OUTARR
002241          DO 3100 M=1,NZ
002243          IR=LOCTHF(M)
002245          INDEX=(M-1)*Nx +L
002261          Y(INDEX)=OUTARR(IR,ICVM)
002266          NPTS(M)=NPTS(M)+1
002260    3100  CONTINUE
002263          GO TO 3070
002263    3200  IF(INCX.EQ.2) GO TO 3250
002265          DO 3230 I=1,Nx
002267          X(I)=XNUM(I)
002271    3230  CONTINUE
002273          GO TO 3340
002273    3250  DO 3280 I=1,Nx
002275          X(I)=I
002277    3280  CONTINUE
002301    3300  IF(ISCALF.EQ.0) GO TO 3320
002302          CALL TYMAXS
002303          GO TO 3350
002304    3320  CALL TIMAXS
002305    3350  CALL PLOT(XLEN+2.5,0.0,-3)
002312          IUNIT=2
002313          IWROTE=1
002314          GO TO 3030
002315    4000  CALL PLOT(XLEN+2.5,0.0,999)
002322          STOP
002324          END
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

                  SUBROUTINE LINEUP
000002            COMMON X(2000),Y(2000),TIME(10),THETA(2,0),OUTARR(21,7),IBUF(1024),
                 1 HEADER(8),NCURVE,NPOINT,YLEN,XLEN,LAB,DT,LUN,LARTYP,XLAB,CONVAR,
                 2 KONTYP,DEL,NPTS(10),ICOL
000002            COMMON/KOFF/ YMIN,DELY,YMINCO,DELYCO,NSPACE,IP,HEIGHT
000002            COMMON/XYPLOT/ XPLOT(200),YPLOT(200),YNUM(5),YMXPOS(6),XCURV(200)
000002            DIF=DT/20.
000004            CVAR=CONVAR+DIF
000005            PVAR=0.
000007       100  IF(CVAR.GE.1.) GO TO 150
000012            CVAR=CVAR*10.
000013            PVAR=PVAR-1.
000015            GO TO 100
000015       150  IF(CVAR.LT.10.) GO TO 175
000020            CVAR=CVAR/10.
000021            PVAR=PVAR+1.
000023            GO TO 150
000024       175  CALL LINES(0.6,0.,-2)
000027            DELX=(X(NPOINT)-X(1))/XLEN
000033            XMIN=X(1)
000034            DO 200 J=1,NPOINT
000035            XPLOT(J)=(X(J)-XMIN)/DELX
000040       200  CONTINUE
000042            NCP=NCURVE*NPOINT
000044            HEIGHT=YLEN/NCURVE -.25
000047            CALL SCALE(Y,HEIGHT,NCP,1)
000052            YMIN=Y(NCP+1)
000054            DELY=Y(NCP+2)
000056            YMAX=HEIGHT*DELY+YMIN
000060            CALL COEF
000061            POW=IP
000063            XL2=XLEN/2.
000065            CALL SYMBOL(XL2-.75,10.75,.15,LAB,0.,16)
000073            CALL SYMBOL(XL2-1.,10.6,.1,6H(IN 10,0.,6)
000101            CALL NUMBER(XL2-0.4,10.65,.07,POW,0.,-1)
000107            CALL SYMBOL(XL2-0.1,10.6,.1,LUN,0.,16)
000115            CALL SYMBOL(XL2+0.3,10.6,.1,1H),0.,1)
000123            CALL SYMBOL(XLEN-1.23,10.6,.07,13HALL CURVES ARE FOR,0.,13)
000131            IF(KONTYP.EQ.2HR=) GO TO 250
000133            IF(ICOL.NE.0.) GO TO 250
000134            CALL GREEK(XLEN+.1,10.6,.1,19,0.,0)
000142            CALL SYMBOL(XLEN+.2,10.6,.1,1H=,0.,1)
000150            GO TO 260
000151       250  CALL SYMBOL(XLEN+.1,10.6,.1,KONTYP,0.,2)
000157       260  CALL NUMBER(XLEN+.3,10.6,.1,CVAR,0.,2)
000165            CALL SYMBOL(XLEN+.7,10.6,.1,3H*10,0.,3)
000173            CALL NUMBER(XLEN+1.,10.65,.07,PVAR,0.,-1)
000201            YSHIFT=-YMIN/DELY
000203            IF(YMIN.GE.0.) YSHIFT=0.0
000206            IF(YMAX.LE.0.) YSHIFT=HEIGHT
000212            DO 300 I=1,NCP
000214            Y(I)=(Y(I)-YMIN)/DELY
000217       300  CONTINUE
000221            ICT=0
000222            YPOS=YLEN+.375-HEIGHT
000225            NP1=NPOINT+1
000227            DO 600 I=1,NCURVE
```

```
000230          DO 400 J=1,NPOINT
000231          ICT=ICT+1
000233          YPLOT(J)=Y(ICT)+YPOS
000236      400 CONTINUE
000240          CALL LINES(XPLOT(1),YPLOT(1),1)
000242          CALL LINES(XPLOT,YPLOT,NPOINT)
000245          YPOPYS=YPOS+YSHIFT
000247          PTIME=0.0
000250          AMPTIM=TIME(I)+DIF
000253      421 IF(AMPTIM.GE.1.) GO TO 460
000256          AMPTIM=AMPTIM*10.
000257          PTIME=PTIME-1.
000261          GO TO 421
000261      460 IF(AMPTIM.LT.10.) GO TO 465
000264          AMPTIM=AMPTIM/10.
000265          PTIME=PTIME+1.
000267          GO TO 460
000270      465 YLAB=YPOPYS+.025
000272          IF(LABTYP.EQ.2HP=) GO TO 466
000274          IF(ICOL.NE.0.) GO TO 466
000275          CALL SYMBOL(XLEN+.1,YLAB,.1,19,0.,0)
000303          CALL SYMBOL(XLEN+.2,YLAB,.1,1H=,0.,1)
000311          GO TO 470
000312      466 CALL SYMBOL(XLEN+.1,YLAB,.1,LABTYP,0.,2)
000320      470 CALL NUMBER(XLEN+.3,YLAB,.1,AMPTIM,0.,2)
000326          CALL SYMBOL(XLEN+.7,YLAB,.1,3H*10,0.,3)
000334          CALL NUMBER(XLEN+1.,YPOPYS+.025,.07,PTIME,0.,-1)
000344          CALL PLOT(XPLOT(NPOINT),YPOPYS,3)
000347          DO 500 K=1,NPOINT
000351          KK=NP1-K
000353          CALL SYMBOL(XPLOT(KK),YPOPYS,.07,13,0.,-2)
000357          CALL PLOT(XPLOT(KK),YPOPYS,3)
000362      500 CONTINUE
000365          YPOS=YPOS-HEIGHT-.20
000370      600 CONTINUE
000372          YINCH=HEIGHT/NSPACE
000374          YSTART=YLEN+.375
000376          YMAX=YMINCH+DELYCH
000400          YINC=DELYCH/NSPACE
000402          NTIC=NSPACE+1
000404          DO 620 I=1,NTIC
000405          YNXPOS(I)=-.35
000407      620 CONTINUE
000411          YNUM(1)=YMAX
000412          IF(YMAX.LT.0.0) YNXPOS(1)=-.42
000415          DO 630 J=2,NTIC
000417          YNUM(J)=YNUM(J-1)-YINC
000422          IF(YNUM(J).LT.0.0) YNXPOS(J)=-.42
000425      630 CONTINUE
000430          DO 700 II=1,NCURVE
000431          CALL PLOT(0.,YSTART,3)
000433          CALL SYMBOL(0.,YSTART,.1,13,90.,-2)
000437          CALL NUMBER(YNXPOS(1),YSTART,.07,YNUM(1),0.,2)
000443          CALL PLOT(0.,YSTART,3)
000446          DO 650 JJ=2,NTIC
000450          YSTART=YSTART-YINCH
000452          CALL SYMBOL(0.,YSTART,.1,13,90.,-2)
000456          CALL NUMBER(YNXPOS(JJ),YSTART,.07,YNUM(JJ),0.,2)
```

```
000463          CALL PLOT(0.,YSTART,3)
000466      650 CONTINUE
000471          YSTART=YSTART-.20
000473      700 CONTINUE
000475          CALL PLOT(XPLOT(NPOINT),YSTART,3)
000500          DO 800 K=1,NPOINT
000502          IF(THETA(K).EQ.90.) X90=XPLOT(K)
000506          KK=NP1-K
000510          CALL SYMBOL(XPLOT(KK),YSTART,.07,13,0.,-2)
000514          CALL PLOT(XPLOT(KK),YSTART,3)
000517      800 CONTINUE
000522          YSTART=YSTART-.07
000524          XLAST=XPLOT(1)
000525          XPNT=XPLOT(NPOINT)
000526          DO 900 J=1,NPOINT
000530          XTEST=XPLOT(J)
000532          IF(J.EQ.1.OR.J.EQ.NPOINT) GO TO 880
000541          IF(XTEST.EQ.X90) GO TO 880
000543          IF(XTEST.GT.X90) GO TO 850
000546          IF(XTEST-XLAST.LT.(.09.OR.X90-XTEST.LT.0.09) GO TO 900
000557          GO TO 880
000560      850 IF(XTEST-XLAST.LT..09.OR.XPNT-XTEST.LT..09) GO TO 900
000572      880 CALL NUMBER(XTEST-.035,YSTART,.07,THETA(J),270.,3)
000601          XLAST=XTEST
000603      900 CONTINUE
000606          CALL SYMBOL(XLEN+.1,YSTART,.07,XLAB,270.,5)
000614          RETURN
000615          END
```

```
                SUBROUTINE LYNEUP
000002          COMMON X(200),Y(2000),TIME(10),THETA(200),OUTARR(21,7),IBUF(1024),
               1 HEADER(8),NCURVE,NPOINT,YLEN,XLEN,LAB,DT,LUN,LABTYP,XLAB,CONVAR,
               2 KONTYP,DEL,NPTS(10),TCOL
000002          COMMON/KOEF/ YMIN,DELY,YMINCO,DELYCO,NSPACE,IP,HEIGHT
000002          COMMON/XYPLOT/ XPLOT(200),YPLOT(200),YNUM(6),YNXPOS(6),XCURV(200)
000002          DTF=DT/2.
000004          CVAR=CONVAR+DTF
000006          PVAR=0.
000007      100 IF(CVAR.GE.1.) GO TO 150
000012          CVAR=CVAR*10.
000013          PVAR=PVAR-1.
000015          GO TO 100
000015      150 IF(CVAR.LT.10.) GO TO 175
000020          CVAR=CVAR/10.
000021          PVAR=PVAR+1.
000023          GO TO 150
000024      175 CALL LINES(0.6,0,-2)
000027          DELX=(X(NPOINT)-X(1))/XLEN
000033          XMIN=X(1)
000034          DO 200 J=1,NPOINT
000035          XPLOT(J)=(X(J)-XMIN)/DELX
000040      200 CONTINUE
000042          NP1=NPOINT+1
000044          HEIGHT=YLEN/NCURVE -.25
000047          XL2=XLEN/2.
000051          YPOS=YLEN+.375-HEIGHT
000054          YSTART=YLEN+.375
000055          IS=0
000056          CALL SYMBOL(XL2-.75,10.75,.15,LAB,0.,10)
000064          CALL SYMBOL(XL2-1.,10.6,.1,6H(IN 10,0.,6)
000072          CALL SYMBOL(XL2-0.4,10.65,.07,3HEXP,0.,3)
000100          CALL SYMBOL(XL2-0.1,10.6,.1,LUN,0.,10)
000106          CALL SYMBOL(XL2+0.9,10.6,.1,1H),0.,1)
000114          CALL SYMBOL(XLEN-1.23,1).6,.07,1HALL CURVES ARE FOR,0.,18)
000122          IF(KONTYP.EQ.2HR=) GO TO 250
000124          IF(TCOL.NE.0.) GO TO 250
000125          CALL GREEK(XLEN+.1,10.6,.1,19,0.,0)
000133          CALL SYMBOL(XLEN+.2,10.6,.1,1H=, .,1)
000141          GO TO 260
000142      250 CALL SYMBOL(XLEN+.1,10.6,.1,KONTYP,0.,2)
000150      260 CALL NUMBER(XLEN+.3,10.6,.1,CVAR,0.,2)
000156          CALL SYMBOL(XLEN+.7,10.6,.1,3H*10,0.,3)
000164          CALL NUMBER(XLEN+1.,10.65,.07,PVAR,0.,-1)
000172          DO 700 II=1,NCURVE
000174          DO 280 J=1,NPOINT
000175          YPLOT(J)=Y(IS+J)
000200      280 CONTINUE
000202          CALL SCALE(YPLOT,HEIGHT,NPOINT,1)
000205          YMIN=YPLOT(NPOINT+1)
000207          DELY=YPLOT(NPOINT+2)
000211          YMAX=HEIGHT*DELY+YMIN
000213          CALL COEF
000214          POW=IP
000216          YSHIFT=-YMIN/DELY
000220          IF(YMIN.GE.0.) YSHIFT=0.0
000222          IF(YMAX.LE.0.) YSHIFT=HEIGHT
```

```
000226          DO 400 J=1,NPOINT
000230          YPLOT(J)=(YPLOT(J)-YMIN)/DELY+YPOS
000234      400 CONTINUE
000237          CALL LINES(XPLOT(1),YPLOT(1),J)
000241          CALL LINES(XPLOT,YPLOT,NPOINT)
000244          YPOPYS=YPOS+YSHIFT
000246          PTIME=0.0
000247          AMPTIM=TIME(II)+DTF
000252      420 IF(AMPTIM.GE.1.) GO TO 450
000255          AMPTIM=AMPTIM*10.
000256          PTIME=PTIME-1.
000260          GO TO 420
000260      450 IF(AMPTIM.LT.10.) GO TO 460
000263          AMPTIM=AMPTIM/10.
000264          PTIME=PTIME+1.
000266          GO TO 450
000267      460 YLAB=YPOPYS+.025
000271          IF(LABTYP.EQ.2HP=) GO TO 465
000273          IF(TOOL.NE.0.) GO TO 465
000274           CALL SPEED(XLEN+.1,YLAB,.1,13,0.,0)
000302          CALL SYMBOL(XLEN+.2,YLAB,.1,IH=,0.,1)
000310          GO TO 470
000311      465 CALL SYMBOL(XLEN+.1,YLAB,.1,LABTYP,0.,2)
000317      470 CALL NUMBER(XLEN+.3,YLAB,.1,AMPTIM,0.,2)
000325          CALL SYMBOL(XLEN+.7,YLAB,.1,3H*10,0.,3)
000333          CALL NUMBER(XLEN+1.,YPOPYS+.075,.07,PTIME,0.,-1)
000343          CALL SYMBOL(XLEN+.35,YPOPYS-.125,.1,4HEXP=,0.,4)
000353          CALL NUMBER(XLEN+.75,YPOPYS-.125,.1,POW,0.,-1)
000363          CALL PLOT(XPLOT(NPOINT),YPOPYS,3)
000366          DO 550 K=1,NPOINT
000370          KK=NP1-K
000372          CALL SYMBOL(XPLOT(KK),YPOPYS,.07,13,0.,-2)
000376          CALL PLOT(XPLOT(KK),YPOPYS,3)
000401      550 CONTINUE
000406          YPOS=YPOS-HEIGHT-.20
000407          YINCH=HEIGHT/NSPACE
000411          YMAX=YMINCO+DELYCO
000413          YINC=DELYCO/NSPACE
000417          NTIC=NSPACE+1
000417          DO 620 I=1,NTIC
000420          YNXPOS(I)=-.35
000422      620 CONTINUE
000424          YNUM(1)=YMAX
000425          IF(YMAX.LT.0.0) YNXPOS(1)=-.42
000430          DO 630 J=2,NTIC
000432          YNUM(J)=YNUM(J-1)-YINC
000435          IF(YNUM(J).LT.0.0) YNXPOS(J)=-.42
000440      630 CONTINUE
000443          CALL PLOT(0.,YSTART,3)
000445          CALL SYMBOL(0.,YSTART,.1,13,90.,-2)
000451          CALL NUMBER(YNXPOS(1),YSTART,.07,YNUM(1),0.,2)
000455          CALL PLOT(0.,YSTART,3)
000460          DO 65) JJ=2,NTIC
000462          YSTART=YSTART-YINCH
000464          CALL SYMBOL(0.,YSTART,.1,13,90.,-2)
000470          CALL NUMBER(YNXPOS(JJ),YSTART,.07,YNUM(JJ),0.,1)
000475          CALL PLOT(0.,YSTART,3)
000500      650 CONTINUE
```

```
000503          YSTART=YSTART-.2
000505          IS=IS+NPOINT
000506      700 CONTINUE
000511          CALL PLOT(XPLOT(NPOINT),YSTART,3)
000514          DO 800 K=1,NPOINT
000516          IF(THETA(K).EQ.90.) X90=XPLOT(K)
000522          KK=NP1-K
000524          CALL SYMBOL(XPLOT(KK),YSTART,.07,13,0.,-2)
000530          CALL PLOT(XPLOT(KK),YSTART,3)
000533      800 CONTINUE
000536          YSTART=YSTART-.07
000540          XLAST=XPLOT(1)
000541          XPNT=XPLOT(NPOINT)
000542          DO 900 J=1,NPOINT
000544          XTEST=XPLOT(J)
000546          IF(J.EQ.1.OR.J.EQ.NPOINT) GO TO 886
000555          IF(XTEST.EQ.X90) GO TO 886
000557          IF(XTEST.GT.X90) GO TO 850
000562          IF(XTEST-XLAST.LT.0.09.OR.X90-XTEST.LT.0.09) GO TO 900
000573          GO TO 886
000574      850 IF(XTEST-XLAST.LT.0.09.OR.XPNT-XTEST.LT.0.09) GO TO 900
000606      886 CALL NUMBER(XTEST-.035,YSTART,.07,THETA(J),270.,3)
000615          XLAST=XTEST
000617      900 CONTINUE
000622          CALL SYMBOL(XLEN+.1,YSTART,.07,XLAB,270.,5)
000630          RETURN
000631          END
```

```
      SUBROUTINE LTCONF
      COMMON X(200),Y(2000),TIME(10),XNUM(200),OUTARR(21,7),IBUF(1024),
     1 HEADER(8),NCURVE,NPOINT,YLEN,XLEN,LAB,DT,LUN,LABTYP,XLAB,CONVAR,
     2 KONTYP,DEL,NPTS(10),ICOL
      COMMON/KOEF/ YMIN,DELY,YMINCO,DELYCO,NSPACE,IP,HEIGHT
      COMMON/XYPLOT/ XPLOT(200),YPLOT(200),YNUM(6),YNXPOS(6),XCURV(200)
      DATA C/3.658/
C
      DTF=DT/2.
      CVAR=CONVAR*DTF
      PVAR=0.
  100 IF(CVAR.GE.1.) GO TO 150
      CVAR=CVAR*10.
      PVAR=PVAR-1.
      GO TO 100
  150 IF(CVAR.LT.10.) GO TO 175
      CVAR=CVAR/10.
      PVAR=PVAR+1.
      GO TO 150
  175 CALL LINES(0.6,0,-2)
      DELX=(X(NPOINT)-X(1))/XLEN
      XMIN=X(1)
      DO 200 J=1,NPOINT
      XPLOT(J)=(X(J)-XMIN)/DELX
  200 CONTINUE
      DELX=DEL/XLEN
      NCP=NCURVE*NPOINT
      HEIGHT=YLEN/NCURVE -.25
      CALL SCALE(Y,HEIGHT,NCP,1)
      YMIN=Y(NCP+1)
      DELY=Y(NCP+2)
      YMAX=HEIGHT*DELY+YMIN
      CALL COEF
      PP=IP
      XL2=XLEN/2.
      CALL SYMBOL(XL2-.75,10.75,.15,LAB,0.,10)
      CALL SYMBOL(XL2-1.,10.6,.1,6HMINUS,0.,6)
      CALL NUMBER(XL2-.4,10.65,.07,POW,0.,-1)
      CALL SYMBOL(XL2-.1,10.6,.1,LUN,0.,10)
      CALL SYMBOL(XL2+.9,10.6,.1,1H),0.,1)
      CALL SYMBOL(XLEN-1.28,10.6,.07,18HALL CURVES ARE FOR,0.,18)
      CALL GOFFK(XLEN+.1,10.6,.1,8,0.,0)
      CALL SYMBOL(XLEN+.2,10.6,.1,1H=,0.,1)
      CALL NUMBER(XLEN+.3,10.6,.1,CVAR,0.,2)
      CALL SYMBOL(XLEN+.7,10.6,.1,3H*10,0.,3)
      CALL NUMBER(XLEN+1.,10.65,.07,PVAR,0.,-1)
      YSHIFT=-YMIN/DELY
      IF(YMIN.GE.0.) YSHIFT=0.0
      IF(YMAX.LE.0.) YSHIFT=HEIGHT
      DO 300 I=1,NCP
      Y(I)=(Y(I)-YMIN)/DELY
  300 CONTINUE
      YPOS=YLEN+.375-HEIGHT
      NPT=NPOINT+1
      DO 400 I=1,NCURVE
      CALL TIMEJ(I)
      L=0
```

```
000223          IN=(I-1)*NPOINT
000226          DO 350 J=1,NPOINT
000230          IF(XNUM(J).GT.CT) GO TO 400
000234          L=L+1
000235      350 CONTINUE
000240      400 NPL=NPOINT-L
000242          XADJ=XMIN+CT-X(L)
000245          JJ=NPL+1
000247          INNPL=IN-NPL
000251          DO 410 J=JJ,NPOINT
000253          IJ=J+INNPL
000255          XCURV(J)=(X(J)-XADJ)/DELX
000260          IF(XCURV(J).LT.0.) XCURV(J)=0.
000262          YPLOT(J)=Y(IJ)+YPOS
000266      410 CONTINUE
000270          CALL LINES(XCURV(JJ),YPLOT(JJ),0)
000273          CALL LINES(XCURV(JJ),YPLOT(JJ),L)
000276          YPOPYS=YPOS+YSHIFT
000300          PTIME=0.0
000301          AMPTIM=TIME(I)+DTF
000304      420 IF(AMPTIM.GE.1.) GO TO 450
000307          AMPTIM=AMPTIM*10.
000310          PTIME=PTIME-1.
000312          GO TO 420
000312      450 IF(AMPTIM.LT.10.) GO TO 460
000315          AMPTIM=AMPTIM/10.
000316          PTIME=PTIME+1.
000320          GO TO 450
000321      460 YLAB=YPOPYS+.225
000323          CALL SYMBOL(XLEN+.1,YLAB,.1,LABTYP,0.,2)
000331          CALL NUMBER(XLEN+.3,YLAB,.1,AMPTIM,0.,2)
000337          CALL SYMBOL(XLEN+.7,YLAB,.1,3H*10,0.,3)
000345          CALL NUMBER(XLEN+1.,YPOPYS+.075,.07,PTIME,0.,-1)
000355          CALL PLOT(XPLOT(NPOINT),YPOPYS,3)
000360          DO 500 K=1,NPOINT
000362          KK=NP1-K
000364          CALL SYMBOL(XPLOT(KK),YPOPYS,.07,13,0.,-2)
000370          CALL PLOT(XPLOT(KK),YPOPYS,3)
000373      500 CONTINUE
000376          YPOS=YPOS-HEIGHT-.20
000401      600 CONTINUE
000403          YINCH=HEIGHT/NSPACE
000405          YSTART=YLEN+.375
000407          YMAX=YMINCO+DELYCO
000411          YINC=DELYCO/NSPACE
000413          NTIC=NSPACE+1
000415          DO 62  I=1,NTIC
000416          YNXPOS(I)=-.35
000420      620 CONTINUE
000422          YNUM(1)=YMAX
000423          IF(YMAX.LT.0.0) YNXPOS(1)=-.42
000426          DO 63  J=2,NTIC
000430          YNUM(J)=YNUM(J-1)-YINC
000433          IF(YNUM(J).LT.0.0) YNXPOS(J)=-.42
000436      630 CONTINUE
000441          DO 700 II=1,NCURVE
000442          CALL PLOT(0.,YSTART,3)
000444          CALL SYMBOL(0.,YSTART,.1,13,90.,-2)
```

```
000450          CALL NUMBER(YNXPOS(1),YSTART,.07,YNUM(1),0.,2)
000454          CALL PLOT(0.,YSTART,3)
000457          DO 650 JJ=2,NTIC
000461          YSTART=YSTART-YINCH
000463          CALL SYMBOL(0.,YSTART,.1,13,90.,-2)
000467          CALL NUMBER(YNXPOS(JJ),YSTART,.07,YNUM(JJ),0.,2)
000474          CALL PLOT(0.,YSTART,3)
000477      650 CONTINUE
000502          YSTART=YSTART-.20
000504      700 CONTINUE
000506          CALL PLOT(XPLOT(NPOINT),YSTART,3)
000511          DO 800 K=1,NPOINT
000513          KK=NPT-K
000515          CALL SYMBOL(XPLOT(KK),YSTART,.07,13,0.,-2)
000521          CALL PLOT(XPLOT(KK),YSTART,3)
000524      800 CONTINUE
000527          YSTART=YSTART-.07
000531          XNORM=XNUM(NPOINT)
000532          XLAST=XPLOT(1)
000533          XPNT=XPLOT(NPOINT)
000535          DO 900 J=1,NPOINT
000536          XNUMM=XNORM-XNUM(J)
000540          XTEST=XPLOT(J)
000542          IF(J.EQ.1.OR.J.EQ.NPOINT) GO TO 880
000561          IF(XTEST-XLAST.LT.0.04.OR.XPNT-XTEST.LT.0.09) GO TO 900
000563      880 CALL NUMBER(XTEST-.035,YSTART,.07,XNUMM,270.,1)
000571          XLAST=XTEST
000573      900 CONTINUE
000576          CALL SYMBOL(XLEN+.1,YSTART,.07,15HDISTANCE INSIDE,0.,15)
000604          CALL SYMBOL(XLEN+.275,YSTART-.105,.07,10HLIGHT CONE,0.,10)
000614          RETURN
000615          END
```

```
                SUBROUTINE LTRONF
000002          COMMON X(200),Y(2000),TIME(10),XNUM(200),OUTARR(21,7),IBUF(1024),
                1 HEADER(8),NCURVE,NPOINT,YLEN,XLEN,LAB,DT,LUN,LABTYP,XLAB,CONVAR,
                2 KONTYP,DEL,NPTS(10),TCUL
000002          COMMON/KOEF/ YMIN,DELY,YMINCO,DELYCO,NSPACE,IP,HEIGHT
000002          COMMON/XYPLOT/ XPLOT(200),YPLOT(200),YNUM(6),YNXPOS(6),XCURV(200)
000002          DATA C/3.0E8/
C
000002          DTF=DT/2.
000004          CVAR=CONVAR*DTF
000006          PVAR=0.
000007      100 IF(CVAR.GE.1.) GO TO 150
000012          CVAR=CVAR*10.
000013          PVAR=PVAR-1.
000015          GO TO 100
000015      150 IF(CVAR.LT.10.) GO TO 175
000020          CVAR=CVAR/10.
000021          PVAR=PVAR+1.
000023          GO TO 150
000024      175 CALL LINES(0,6,0,-2)
000027          DELX=(X(NPOINT)-X(1))/XLEN
000033          XMIN=X(1)
000034          DO 200 J=1,NPOINT
000035          XPLOT(J)=(X(J)-XMIN)/DELX
000040      200 CONTINUE
000042          DELX=DEL/XLEN
000044          NP1=NPOINT+1
000046          HEIGHT=YLEN/NCURVE  -.25
000051          XL2=XLEN/2.
000053          YPOS=YLEN+.375-HEIGHT
000056          YSTART=YLEN+.375
000057          IS=0
000060          CALL SYMBOL(XL2-.75,10.75,.15,LAB,0.,10)
000066          CALL SYMBOL(XL2-1.,10.6,.1,6H(IN 10,0.,6)
000074          CALL SYMBOL(XL2-0.4,10.65,.07,3HEXP,0.,3)
000102          CALL SYMBOL(XL2-0.1,10.6,.1,LUN,0.,10)
000110          CALL SYMBOL(XL2+0.9,10.6,.1,1H),0.,1)
000116          CALL SYMBOL(XLEN-1.28,10.6,.07,18HALL CURVES ARE FOR,0.,18)
000124          CALL GREEK(XLEN+.1,10.6,.1,8,0.,0)
000132          CALL SYMBOL(XLEN+.2,10.6,.1,1H=,0.,1)
000140          CALL NUMBER(XLEN+.3,10.6,.1,CVAR,0.,2)
000146          CALL SYMBOL(XLEN+.7,10.6,.1,3H*10,0.,3)
000154          CALL NUMBER(XLEN+1.,10.65,.07,PVAR,0.,-1)
000162          DO 700 II=1,NCURVE
000164          DO 250 J=1,NPOINT
000165          YPLOT(J)=Y(IS+J)
000170      250 CONTINUE
000172          CALL SCALE(YPLOT,HEIGHT,NPOINT,1)
000175          YMIN=YPLOT(NPOINT+1)
000177          DELY=YPLOT(NPOINT+2)
000201          YMAX=HEIGHT*DELY+YMIN
000203          CALL COEF
000204          POW=IP
000206          YSHIFT=-YMIN/DELY
000210          IF(YMIN.GE.0.) YSHIFT=0.0
000212          IF(YMAX.LE.0.) YSHIFT=HEIGHT
000216          CT=C*TIME(II)
```

```
000221          L=0
000222          DO 300 J=1,NPOINT
000223          IF(XNUM(J).GT.CT) GO TO 350
000227          L=L+1
000230      300 CONTINUE
000232      350 NPL=NPOINT-L
000234          XADJ=XMIN+CT-X(L)
000237          J1=NPL+1
000241          NPJ=NPOINT+J1
000243          NPJL=NPJ-NPL
000244          DO 400 J=J1,NPOINT
000245          IJ=NPJ-J
000247          IJL=NPJL-J
000250          XCURV(J)=(X(J)-XADJ)/DELX
000263          IF(XCURV(J).LT.0.) XCURV(J)=0.
000266          YPLOT(IJ)=(YPLOT(IJL)-YMIN)/DELY+YPOS
000243      400 CONTINUE
000266          CALL LINES(XCURV(J1),YPLOT(J1),0)
000271          CALL LINES(XCURV(J1),YPLOT(J1),L)
000274          YPOPYS=YPOS+YSHIFT
000276          PTIME=0.0
000277          AMPTIM=TIME(II)+DTF
000302      420 IF(AMPTIM.GE.1.) GO TO 450
000305          AMPTIM=AMPTIM*10.
000306          PTIME=PTIME-1.
000310          GO TO 420
000310      450 IF(AMPTIM.LT.10.) GO TO 460
000313          AMPTIM=AMPTIM/10.
000314          PTIME=PTIME+1.
000316          GO TO 450
000317      460 YLAB=YPOPYS+.025
000321          CALL SYMBOL(XLEN+.1,YLAB,.1,LABTYP,0.,2)
000327          CALL NUMBER(XLEN+.3,YLAB,.1,AMPTIM,0.,2)
000335          CALL SYMBOL(XLEN+.7,YLAB,.1,3H*10,0.,3)
000343          CALL NUMBER(XLEN+1.,YPOPYS+.075,.07,PTIME,0.,-1)
000353          CALL SYMBOL(XLEN+.35,YPOPYS-.125,.1,4HEXP=,0.,4)
000363          CALL NUMBER(XLEN+.75,YPOPYS-.125,.1,POW,0.,-1)
000373          CALL PLOT(XPLOT(NPOINT),YPOPYS,3)
000376          DO 500 K=1,NPOINT
000400          KK=NP1-K
000402          CALL SYMBOL(XPLOT(KK),YPOPYS,.07,13,0.,-2)
000426          CALL PLOT(XPLOT(KK),YPOPYS,3)
000411      500 CONTINUE
000414          YPOS=YPOS-HEIGHT-.20
000417          YINCH=HEIGHT/NSPACE
000421          YMAX=YMINCO+DELYCO
000423          YINC=DELYCO/NSPACE
000425          NTIC=NSPACE+1
000427          DO 620 I=1,NTIC
000430          YNXPOS(I)=-.35
000432      620 CONTINUE
000434          YNUM(1)=YMAX
000435          IF(YMAX.LT.0.0) YNXPOS(1)=-.42
000440          DO 630 J=2,NTIC
000442          YNUM(J)=YNUM(J-1)-YINC
000445          IF(YNUM(J).LT.0.0) YNXPOS(J)=-.42
000450      630 CONTINUE
000453          CALL PLOT(0.,YSTART,3)
```

```
000455          CALL SYMBOL(0.,YSTART,.1,13,90.,-2)
000461          CALL NUMBER(YNXPOS(1),YSTART,.07,YNUM(1),0.,2)
000465          CALL PLOT(0.,YSTART,3)
000470          DO 650 JJ=2,NTIC
000472          YSTART=YSTART-YINCH
000474          CALL SYMBOL(0.,YSTART,.1,13,90.,-2)
000500          CALL NUMBER(YNXPOS(JJ),YSTART,.07,YNUM(JJ),0.,2)
000505          CALL PLOT(0.,YSTART,3)
000510      650 CONTINUE
000513          YSTART=YSTART-.20
000515          IS=IS+NPOINT
000516      700 CONTINUE
000521          CALL PLOT(XPLOT(NPOINT),YSTART,3)
000524          DO 800 K=1,NPOINT
000526          KK=NPT-K
000530          CALL SYMBOL(XPLOT(KK),YSTART,.07,13,0.,-2)
000534          CALL PLOT(XPLOT(KK),YSTART,3)
000537      800 CONTINUE
000542          YSTART=YSTART-.07
000544          XNORM=XNUM(NPOINT)
000545          XLAST=XPLOT(1)
000546          XPNT=XPLOT(NPOINT)
000550          DO 900 J=1,NPOINT
000561          XNUMM=XNORM-XNUM(J)
000563          XTEST=XPLOT(J)
000555          IF(J.EQ.1.OR.J.EQ.NPOINT) GO TO 880
000564          IF(XTEST-XLAST.LT.0.09.OR.XPNT-XTEST.LT.0.09) GO TO 900
000576      880 CALL NUMBER(XTEST-.035,YSTART,.07,XNUMM,270.,1)
000604          XLAST=XTEST
000606      900 CONTINUE
000611          CALL SYMBOL(XLEN+.1,YSTART-.07,.15HDISTANCE INSIDE,0.,15)
000617          CALL SYMBOL(XLEN+.275,YSTART-.105,.07,16HLIGHT CONE,0.,16)
000627          RETURN
000630          END
```

```
      SUBROUTINE TIMAXS
000002      COMMON X(200),Y(2000),Z(10),XNUM(200),OUTARR(21,7),IBUF(1024),
     1 HEADER(8),NCURVE,NPOINT,YLEN,XLEN,LAB,DT,LUN,LABTYP,XLAB,CONVAR,
     2 KONTYP,NEL,NPTS(10),ICOL
000002      COMMON/KOEF/ YMIN,DELY,YMINCO,DELYCO,NSPACE,IP,HEIGHT
000002      COMMON/XYPLOT/ XPLOT(200),YPLOT(200),YNUM(6),YNXPOS(6),XCURV(200)
C
000002      DTF=DT/20.
000004      CVAR=CONVAR+DTF
000006      PVAR=0.
000007  100 IF(CVAR.GE.1.) GO TO 150
000012      CVAR=CVAR*10.
000013      PVAR=PVAR-1.
000015      GO TO 100
000015  150 IF(CVAR.LT.10.) GO TO 175
000020      CVAR=CVAR/10.
000021      PVAR=PVAR+1.
000023      GO TO 150
000024  175 CALL LINES(0.6,0,-2)
000027      DELX=(X(NPOINT)-X(1))/XLEN
000033      XMIN=X(1)
000034      DO 200 J=1,NPOINT
000035      XPLOT(J)=(X(J)-XMIN)/DELX
000040  200 CONTINUE
000042      NCP=NCURVE*NPOINT
000044      HEIGHT=YLEN/NCURVE -.25
000047      YMAX=Y(1)
000051      YMIN=Y(1)
000052      DO 250 I=1,NCURVE
000053      NPT=NPTS(I)
000055      ICT=(I-1)*NPOINT
000060      DO 240 J=1,NPT
000062      ICT=ICT+1
000064      YPLOT(J)=Y(ICT)
000066  240 CONTINUE
000070      CALL SCALE(YPLOT,HEIGHT,NPT,1)
000073      YTEST=YPLOT(NPT+2)*HEIGHT+YPLOT(NPT+1)
000077      IF(YMAX.LT.YTEST) YMAX=YTEST
000102      IF(YMIN.GT.YPLOT(NPT+1)) YMIN=YPLOT(NPT+1)
000106  250 CONTINUE
000111      DELY=(YMAX-YMIN)/HEIGHT
000114      CALL COEF
000115      PO=IP
000117      XL2=XLEN/2.
000121      CALL SYMBOL(XL2-.75,1.,.75,.15,LAB,0.,10)
000127      CALL SYMBOL(XL2-1.,10.6,.1,6H(IN 10,0.,6)
000135      CALL NUMBER(XL2-0.4,10.65,.07,POW,0.,-1)
000143      CALL SYMBOL(XL2-0.1,10.6,.1,LUN,0.,1)
000151      CALL SYMBOL(XL2+0.9,10.6,.1,1H),0.,1)
000157      IF(KONTYP.EQ.2H  ) GO TO 260
000161      CALL SYMBOL(XLEN-1.23,10.6,.07,18HALL CURVES ARE FOR,0.,18)
000167      CALL SYMBOL(XLEN+.1,10.6,.1,KONTYP,0.,2)
000175      GO TO 270
000176  260 CALL SYMBOL(XLEN-1.28,10.6,.07,18HALL CURVES ARE FOR,0.,18)
000204      CALL GREEK(XLEN+.1,10.6,.1,8,0.,0)
000212      CALL SYMBOL(XLEN+.2,10.6,.1,1H=,0.,1)
000220  270 CALL NUMBER(XLEN+.3,10.6,.1,CVAR,0.,2)
```

```
000226          CALL SYMROL(XLEN+.7,10.6,.1,3H*10,0.,3)
000234          CALL NUMBER(XLEN+1.,10.65,.07,PVAR,0.,-1)
000242          YSHIFT=-YMIN/DELY
000244          IF(YMIN.GE.0.) YSHIFT=0.0
000247          IF(YMAX.LE.0.) YSHIFT=HEIGHT
000253          DO 300 I=1,NCP
000255          Y(I)=(Y(I)-YMIN)/DELY
000260      300 CONTINUE
000262          ICT=0
000263          YPOS=YLEN+.375-HEIGHT
000266          NPT=NPOINT+1
000270          DO 600 I=1,NCURVE
000271          ICT=(I-1)*NPOINT
000274          NPT=NPTS(I)
000276          DO 400 J=1,NPT
000277          ICT=ICT+1
000301          XPLOT(J)=Y(ICT)+YPOS
000304      400 CONTINUE
000306          CALL LINES(XPLOT(1),YPLOT(1),0)
000310          CALL LINES(XPLOT,YPLOT,NPT)
000313          YPOPYS=YPOS+YSHIFT
000315          PZ=0.
000316          AMPZ=Z(I)+DTF
000321      420 IF(AMPZ.GE.1.) GO TO 450
000324          AMPZ=AMPZ*10.
000325          PZ=PZ-1.
000327          GO TO 420
000327      450 IF(AMPZ.LT.10.) GO TO 460
000332          AMPZ=AMPZ/10.
000333          PZ=PZ+1.
000335          GO TO 450
000336      460 YLAB=YPOPYS+.225
000340          IF(LABTYP.EQ.2H  ) GO TO 470
000342          CALL SYMROL(XLEN+.1,YLAB,.1,LABTYP,0.,2)
000350          GO TO 480
000351      470 CALL GREEK(XLEN+.1,YLAB,.1,8,0.,0)
000357          CALL SYMROL(XLEN+.2,YLAB,.1,1H=,0.,1)
000365      480 CALL NUMBER(XLEN+.3,YLAB,.1,AMPZ,0.,2)
000373          CALL SYMROL(XLEN+.7,YLAB,.1,3H*10,0.,3)
000401          CALL NUMBER(XLEN+1.,YPOPYS+.075,.07,PZ,0.,-1)
000411          CALL PLOT(XPLOT(NPOINT),YPOPYS,3)
000414          DO 500 K=1,NPOINT
000416          KK=NPT-K
000420          CALL SYMROL(XPLOT(KK),YPOPYS,.07,13,0.,-2)
000424          CALL PLOT(XPLOT(KK),YPOPYS,3)
000427      500 CONTINUE
000432          YPOS=YPOS-HEIGHT-.20
000435      600 CONTINUE
000437          YINCH=HEIGHT/NSPACE
000441          YSTART=YLEN+.375
000443          YMAX=YMINCO+DELYCO
000445          YINC=DELYCO/NSPACE
000447          NTIC=NSPACE+1
000451          DO 620 I=1,NTIC
000452          YNXPOS(I)=-.35
000454      620 CONTINUE
000456          YNUM(1)=YMAX
000457          IF(YMAX.LT.0.0) YNXPOS(1)=-.42
```

```
000442          DO 63  J=2,NTIC
000444          YNUM(J)=YNUM(J-1)-YINC
000447          IF(YNUM(J).LT.0.0) YNXPOS(J)=-.42
000472      630 CONTINUE
000475          DO 700 II=1,NCURVE
000476          CALL PLOT(0.,YSTART,3)
000500          CALL SYMBOL(0.,YSTART,.1,13,90.,-2)
000504          CALL NUMBER(YNXPOS(1),YSTART,.07,YNUM(1),0.,2)
000510          CALL PLOT(0.,YSTART,3)
000513          DO 650 JJ=2,NTIC
000515          YSTART=YSTART-YINCH
000517          CALL SYMBOL(0.,YSTART,.1,13,90.,-2)
000523          CALL NUMBER(YNXPOS(JJ),YSTART,.07,YNUM(JJ),0.,2)
000530          CALL PLOT(0.,YSTART,3)
000533      650 CONTINUE
000536          YSTART=YSTART-.25
000540      700 CONTINUE
000542          CALL PLOT(XPLOT(NPOINT),YSTART,3)
000545          DO 800 K=1,NPOINT
000547          KK=NPT-K
000551          CALL SYMBOL(XPLOT(KK),YSTART,.07,13,0.,-2)
000555          CALL PLOT(XPLOT(KK),YSTART,3)
000560      800 CONTINUE
000563          YSTART=YSTART-.07
000565          XLAST=XPLOT(1)
000566          XPNT=XPLOT(NPOINT)
000567          DO 900 J=1,NPOINT
000571          XNUMM=XNUM(J)+1.E9
000573          XTEST=XPLOT(J)
000575          IF(J.EQ.1.OR.J.EQ.NPOINT) GO TO 880
000604          IF(XTEST-XLAST.LT.0.03.OR.XPNT-XTEST.LT.0.09) GO TO 900
000616      880 CALL NUMBER(XTEST-.035,YSTART,.07,XNUMM,270.,1)
000624          XLAST=XTEST
000626      900 CONTINUE
000631          IF(TCOL.EQ.0.) GO TO 1000
000632          CALL SYMBOL(XLEN+.1,YSTART,.07,9HT IN NSEC,0.,9)
000640          RETURN
000641     1000 CALL GREEK(XLEN+.1,YSTART,.1,19,0.,0)
000647          CALL SYMBOL(XLEN+.3,YSTART,.07,7HIN NSEC,0.,7)
000655          RETURN
000666          END
```

```
                      SUBROUTINE TYMAXS
000002                COMMON X(2000),Y(2000),Z(10),XNUM(200),OUTARR(21,7),IBUF(1024),
                     1 HEADER(8),NCURVE,NPOINT,YLEN,XLEN,LAB,DT,LUN,LABTYP,XLAB,CONVAR,
                     2 KONTYP,DEL,NPTS(10),TCUL
000002                COMMON/KOEF/ YMIN,DELY,YMINCO,DELYCO,NSPACE,IP,HEIGHT
000002                COMMON/XYPLOT/ XPLOT(200),YPLOT(200),YNUM(6),YNXPOS(6),XCURV(200)
                     C
000002                DTF=DT/2.
000004                CVAR=CONVAR*DTF
000006                PVAR=1.
000007           100  IF(CVAR.GE.1.) GO TO 150
000012                CVAR=CVAR*10.
000013                PVAR=PVAR-1.
000015                GO TO 100
000015           150  IF(CVAR.LT.10.) GO TO 175
000020                CVAR=CVAR/10.
000021                PVAR=PVAR+1.
000023                GO TO 150
000024           175  CALL LINES(0.6,0,-2)
000027                DELX=(X(NPOINT)-X(1))/XLEN
000033                XMIN=X(1)
000034                DO 200 J=1,NPOINT
000036                XPLOT(J)=(X(J)-XMIN)/DELX
000040           200  CONTINUE
000042                NP1=NPOINT+1
000044                HEIGHT=YLEN/NCURVE *.25
000047                XL2=XLEN/2.
000051                YPOS=YLEN+.375-HEIGHT
000054                YSTART=YLEN+.375
000055                CALL SYMBOL(XL2-.75,1 .75,.15,LAB,0.,10)
000063                CALL SYMBOL(XL2-1.,10.6,.1,6HFIN 10,0.,6)
000071                CALL SYMBOL(XL2-0.4,10.65,.07,3HEXP,0.,3)
000077                CALL SYMBOL(XL2-0.1,10.6,.1,LUN,0.,10)
000105                CALL SYMBOL(XL2+0.9,10.6,.1,1H),0.,1)
000113                IF(KONTYP.EQ.2H  ) GO TO 220
000114                CALL SYMBOL(XLEN-1.23,10.6,.07,18HALL CURVES ARE FOR,0.,18)
000123                CALL SYMBOL(XLEN+.1,1 .6,.1,KONTYP,0.,2)
000131                GO TO 240
000132           220  CALL SYMBOL(XLEN-1.28,10.6,.07,18HALL CURVES ARE FOR,0.,13)
000140                CALL GREFK(XLEN+.1,10.6,.1,8,0.,0)
000146                CALL SYMBOL(XLEN+.2,10.6,.1,1H=,0.,1)
000154           240  CALL NUMBER(XLEN+.3,10.6,.1,CVAR,0.,2)
000162                CALL SYMBOL(XLEN+.7,10.6,.1,3H*10,0.,3)
000170                CALL NUMBER(XLEN+1.,10.65,.07,PVAR,0.,-1)
000176                DO 700 II=1,NCURVE
000200                NPT=NPTS(II)
000202                ICT=(II-1)*NPOINT
000205                DO 250 J=1,NPT
000207                YPLOT(J)=Y(ICT+J)
000212           250  CONTINUE
000214                CALL SCALE(YPLOT,HEIGHT,NPT,1)
000217                YMIN=YPLOT(NPT+1)
000221                DELY=YPLOT(NPT+2)
000223                YMAX=HEIGHT*DELY+YMIN
000225                CALL COEF
000226                POW=IP
000230                YSHIFT=-YMIN/DELY
```

```
000232          IF(YMIN.GE.0.) YSHIFT=0.0
000234          IF(YMAX.LE.0.) YSHIFT=HEIGHT
000240          DO 400 J=1,NPT
000242          YPLOT(J)=(YPLOT(J)-YMIN)/DELY +YPOS
000246      400 CONTINUE
000251          CALL LINES(XPLOT(1),YPLOT(1),0)
000253          CALL LINES(XPLOT,YPLOT,NPT)
000256          YPOPYS=YPOS+YSHIFT
000260          PZ=0.0
000261          AMPZ=Z(II)+DTF
000264      420 IF(AMPZ.GE.1.) GO TO 450
000267          AMPZ=AMPZ*10.
000270          PZ=PZ-1.
000272          GO TO 420
000272      450 IF(AMPZ.LT.10.) GO TO 460
000275          AMPZ=AMPZ/10.
000276          PZ=PZ+1.
000300          GO TO 450
000301      460 YLAB=YPOPYS+.225
000303          IF(LABTYP.EQ.5H    ) GO TO 470
000305          CALL SYMBOL(XLEN+.1,YLAB,.1,LABTYP,0.,2)
000313          GO TO 480
000314      470 CALL GREEK(XLEN+.1,YLAB,.1,8,0.,0)
000322          CALL SYMBOL(XLEN+.2,YLAB,.1,1H=,0.,1)
000330      480 CALL NUMBER(XLEN+.3,YLAB,.1,AMPZ,0.,2)
000336          CALL SYMBOL(XLEN+.7,YLAB,.1,3H*10,0.,3)
000344          CALL NUMBER(XLEN+1.,YPOPYS+.075,.07,PZ,0.,-1)
000354          CALL SYMBOL(XLEN+.35,YPOPYS-.125,.1,4HEXP=,0.,4)
000364          CALL NUMBER(XLEN+.75,YPOPYS-.125,.1,POW,0.,-1)
000374          CALL PLOT(XPLOT(NPOINT),YPOPYS,3)
000377          DO 500 K=1,NPOINT
000401          KK=NPT-K
000403          CALL SYMBOL(XPLOT(KK),YPOPYS,.07,13,0.,-2)
000407          CALL PLOT(XPLOT(KK),YPOPYS,3)
000412      500 CONTINUE
000415          YPOS=YPOS-HEIGHT-.20
000420          YINCH=HEIGHT/NSPACE
000422          YMAX=YMINCO+DELYCO
000424          YINC=DELYCO/NSPACE
000426          NTIC=NSPACE+1
000430          DO 620 I=1,NTIC
000431          YNXPOS(I)=-.36
000433      620 CONTINUE
000435          YNUM(1)=YMAX
000436          IF(YMAX.LT.0.0) YNXPOS(1)=-.42
000441          DO 630 J=2,NTIC
000443          YNUM(J)=YNUM(J-1)-YINC
000446          IF(YNUM(J).LT.0.0) YNXPOS(J)=-.42
000451      630 CONTINUE
000454          CALL PLOT(0.,YSTART,3)
000456          CALL SYMBOL(0.,YSTART,.1,13,90.,-2)
000462          CALL NUMBER(YNXPOS(1),YSTART,.07,YNUM(1),0.,2)
000466          CALL PLOT(0.,YSTART,3)
000471          DO 65  JJ=2,NTIC
000473          YSTART=YSTART-YINCH
000475          CALL SYMBOL(0.,YSTART,.1,13,90.,-2)
000501          CALL NUMBER(YNXPOS(JJ),YSTART,.07,YNUM(JJ),0.,2)
000506          CALL PLOT(0.,YSTART,3)
```

```
000511      650 CONTINUE
000514          YSTART=YSTART-.25
000516      700 CONTINUE
000520          CALL PLOT(XPLOT(NPOINT),YSTART,3)
000523          DO 800 K=1,NPOINT
000525          KK=NP1-K
000527          CALL SYMBOL(XPLOT(KK),YSTART,.07,13,0.,-2)
000533          CALL PLOT(XPLOT(KK),YSTART,3)
000536      800 CONTINUE
000541          YSTART=YSTART-.07
000543          XLAST=XPLOT(1)
000544          XPNT=XPLOT(NPOINT)
000545          DO 900 J=1,NPOINT
000547          XNUMM=XNUM(J)*1.E9
000561          XTEST=XPLOT(J)
000563          IF(J.EQ.1.OR.J.EQ.NPOINT) GO TO 880
000565          IF(XTEST-XLAST.LT.0.09.OR.XPNT-XTEST.LT.0.09) GO TO 900
000574      880 CALL NUMBER(XTEST-.035,YSTART,.07,XNUMM,270.,-1)
000602          XLAST=XTEST
000604      900 CONTINUE
000607          IF(TCOL.EQ.0.) GO TO 1000
000610          CALL SYMBOL(XLEN+.1,YSTART,.07,9HT IN NSEC,0.,9)
000616          RETURN
000617     1000 CALL GREEK(XLEN+.1,YSTART,.1,19,0.,0)
000625          CALL SYMBOL(XLEN+.3,YSTART,.07,7HIN NSEC,0.,7)
000633          RETURN
000634          END
```

```
                SUBROUTINE COFF
000002          COMMON/KOEF/ YMIN,DELY,YMINC,DELYC,NRET,I,HEIGHT
000002          DIMENSION YAMP(10),NT(10)
000002          DATA YAMP/1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0/
000002          DATA NT/5,5,5,4,5,4,5,4,5,5/
000002          I=0
000003          DELYC=DELY*HEIGHT
000005      100 IF(DELYC.LT.10.) GO TO 200
000010          DELYC=DELYC/10.
000011          I=I+1
000013          GO TO 100
000014      200 IF(DELYC.GE.1.) GO TO 300
000017          DELYC=DELYC*10.
000020          I=I-1
000022          GO TO 200
000023      300 YMINC=YMIN*10.0**(-I)
000030          DELYC=HEIGHT*DELY*10.0**(-I)
000040          DO 400 K=1,9
000041          IF(DELYC.LE.YAMP(K)) GO TO 500
000044      400 CONTINUE
000045          K=10
000046      500 NRET=NT(K)
000050          RETURN
000051          END
```

```
      SUBROUTINE LINES(X,Y,KZ)
      DIMENSION X(1),Y(1),ZEND(5),JK(5)
000006 DIMENSION NJJ(10),XZ(10),XZEND(25),NJK(24),LOC(10)
000006 EQUIVALENCE(INTENS,XINT)
000006 DATA NJJ/1,1,2,1,2,3,4,3,4,4/,LOC/0,1,2,4,5,7,10,14,17,21/
000006 DATA XZ/1000.,0.08,0.1,0.0,0.0,0.15,0.15,0.2,0.2,0.25,0.25/,NJK/1,3,
     [1,2,3,1,2,1,2,3,1,2,3,4,1,2,3,1,2,3,4,1,2,1,2/,XZEND/100.,0.08,0.0
     [5,0.1,0.040,0.1,0.15,0.05,0.1,0.15,0.05,0.1,0.15,0.2,0.1,0.15,0.2,
     [0.1,0.15,0.2,0.25,0.1,0.15,0.2,0.25/
000006  IF(KZ.GT.0) GO TO 105
000010  IF(KZ.EQ.0) GO TO 40
000010  XINT=Y(1)
000011  SCALE=X(1)
000012  II=-KZ
000013  RETURN
000014 40 IF(II.GT.1) GO TO 50
000020  CALL PLOT(X(1),Y(1),3)
000022  RETURN
000023 50 CONTINUE
000023  JJ=NJJ(II)
000025  Z=XZ(II)*SCALE
000027  DO 100 I=1,JJ
000033  IJ=I+LOC(II)
000035  ZEND(I)=XZEND(IJ)*SCALE
000040  IF(NJK(IJ).NE.1) GO TO 60
000042  ASSIGN 140 TO J
000043  GO TO 90
000043 60 IF(NJK(IJ).NE.2) GO TO 70
000046  ASSIGN 130 TO J
000047  GO TO 90
000047 70 IF(NJK(IJ).NE.3) GO TO 80
000052  ASSIGN 150 TO J
000053  GO TO 90
000053 80 ASSIGN 160 TO J
000054 90 JK(I)=J
000056 100 CONTINUE
000061  IJ=1
000062  ZLEFT=Z
000063  JDOT1=5
000064  JDOT2=2
000065  OLDX=X(1)
000066  OLDY=Y(1)
000067  RETURN
000067 105 CONTINUE
000067  NPTS=KZ
000070  IF(II.LE.1) GO TO 210
000073  DO 200 I=1,NPTS
000074  XB=OLDX
000076  YB=OLDY
000077  OLDX=X(I)
000101  OLDY=Y(I)
000102  S=SQRT((X(I)-XB)**2+(Y(I)-YB)**2)
000114  IF (S.EQ.0.0) GO TO 200
000115  XFACT=(X(I)-XB)/S
000120  YFACT=(Y(I)-YB)/S
000123 111 CONTINUE
000123  XA=XB
```

```
000125          YA=YB
000126    110   SX=AMIN1(S,-Z+ZLEFT+ZEND(IJ))
000134          IF(SX.GT.0.0) GO TO 120
000136          IJ=IJ+1
000137          IF(IJ.LE.JJ) GO TO 110
000141          IJ=1
000142          ZLEFT=Z
000143          JDOT1=0
000144          JDOT2=0
000145          GO TO 110
000145    120   CONTINUE
000145          XB=XA+SX*XFACT
000150          YB=YA+SX*YFACT
000153          JJJ=JK(IJ)
000155          GO TO JJJ,(130,140,150,160)
000160    130   CONTINUE
000160          S=S-SX
000162          ZLEFT=ZLEFT-SX
000164          IF(S.LE.0.0) GO TO 200
000165          GO TO 110
000165    140   CONTINUE
000165          CALL PLOT (XA,YA,3)
000170          CALL PLOT (XB,YB,2)
000173          GO TO 130
000176    150   CONTINUE
000176          IF(JDOT1.NE.0) GO TO 130
000177          CALL PLOT (XA,YA,3)
000202          CALL PLOT (XA,YA,2)
000205          JDOT1=1
000206          GO TO 130
000211    160   CONTINUE
000211          IF(JDOT2.NE.0) GO TO 130
000212          CALL PLOT (XA,YA,3)
000215          CALL PLOT (XA,YA,2)
000220          JDOT2=1
000221          GO TO 130
000224    200   CONTINUE
000227          RETURN
000227    210   CONTINUE
000227          DO 220 I=1,NPTS
000231          CALL PLOT (X(I),Y(I),2)
000235    220   CONTINUE
000242          RETURN
000242          END
*FOLLOWING VARIABLES EQUIVALENCED BUT NOT REFERENCED
INTENS
```

```
                SUBROUTINE SCALE(ARRAY,AXLEN,NPTS,IMIX)
000007          DIMENSION ARRAY(1)
000007          AMAX=ARRAY(1)
000007          AMIN=AMAX
000011          DO 1 0 I=2,NPTS
000013          AMAX=AMAX1(AMAX,ARRAY(I))
000017          AMIN=AMIN1(AMIN,ARRAY(I))
000023      100 CONTINUE
000025          ARRAY(NPTS+1)=AMIN
000026          IF(AMAX.EQ.AMIN) GO TO 200
000030          ARRAY(NPTS+2)=(AMAX-AMIN)/AXLEN
000032          RETURN
000033      200 ARRAY(NPTS+2)=ABS(2.0*AMIN/AXLEN)+1
000041          RETURN
000042          END
```

```
C     SUBROUTINE GREEK(XPOS,YPOS,SCALE,LETTER,THETA,ICAP)
C
C                          LETTER CODES
C
C          1 ALPHA        7 ETA         13 NU            19 TAU
C          2 BETA         8 THETA       14 XI            20 UPSILON
C          3 GAMMA        9 IOTA        15 OMICRON       21 PHI
C          4 DELTA       10 KAPPA       16 PI            22 CHI
C          5 EPSILON     11 LAMBDA      17 RHO           23 PSI
C          6 ZETA        12 MU          18 SIGMA         24 OMEGA
C
C          LOWER CASE -- ICAP=0         UPPER CASE -- ICAP=1
C
000011      DIMENSION XYINCH(32),IPOINT(159),JPOINT(56),KPOINT(47),ISTART(24)
000011      EQUIVALENCE (IPOINT(57),JPOINT(1)),(IPOINT(113),KPOINT(1))
000011      DATA XYINCH/
           1 .03125,.0625,.09375,.125,.15625,.1875,.21875,.25,.28125,.3125,
           2 .34375,.375,.40625,.4375,.46875,.5,.53125,.5625,.59375,.625,
           3 .65625,.6875,.71875,.75,.78125,.8125,.84375,.875,.90625,.9375,
           4 .96875,1.0/
000011      DATA IPOINT/
           1 3125235221461642134008,1040054102440051005558,0262056510661366166408
           2,2157234525412740304048,1420000000000000000008,0000036705731076137708
           3,2377227623742371226608,1763136105611321166008,2057225523532350214508
           4,1644124404464151000008,2533217115641461135208,1340145214611267107208
           5,0573 3734070000000008,2333207615771277176008,0773107020602255235108
           6,2245204214400640024348,0050005602630565516708,2127126604640161005408
           7,0051014460443124121408,0015575500000000000008,0037037517752076207708
           8,1577117506720365005508,0046024520462144174208,5040000000000000000008
           9,0031017203720470024608,0425077113721672227008,2465246262400000000008
           A,0013006401710375067708,1277157517712064205308,1746154212400640034208
           B,0146005300206060000008,0525034704430641104408,1240144156440000000008
           C,0326004020260255604008,0135037705762241244008,2642122240400000000008
           D,0432025200400212044708,0746134616472052227208,2012204721462246445008
           E,0012703670460045203408,0440124417522260236308,6367000000000000000008
           F,1037107512742074200508,1175057303710266036308,0061206106210356015308
           G,0050014504442043204108,1640524000000000000008,1327166620642261035508
           H,2352224620431641134008,1040054103430146005208,0055016103640565106708
           I,5367000000000000000008,3127306002660165006308,0061072607402226424008
           J,0000036404700673127548,1675217423722467246408/
000011      DATA JPOINT/
           1 2260165512550756040148,4364000000000000000008,2531237016671367106708
           2,0566036401610055005248,0146034305411 4013408,1641204322462352235508
           3,2261206416665367000008,2627036702600165006308,0062122712421341144008
           4,0040000000000000000008,0027036704660464036 8,0350054310401440174108
           5,2144235123602264616708,1427206622652463206008,2557245422522051145008
           6,1150055103520154005708,0067016303650566116708,1461153750400000000008
           7,0032017304730572174108,2040234024412333404408,0031027003660363055408
           8,0751115013501652226048,2470257115375140000008,0527026401620056005208
           9,0146034206401040134248,1444145714041542204 8,2240254227463052305608
           A,2762266463670000000008,0000044002001477244 8,2400304006146254000008
           B,0 3721772476267427714B,2766266324612160026 8,2120245726552752274608
           C,2643244121400040023748,4240000000000000000 8,2632267700770237024008
           D,0000444000000000000008,0000147730434340000 8,2732277700770237024008
           E,0000274027450220176008,1723575500000000000008,0032007725770040254008
           F,6545 0000000000000008,0037047702370240000 8,0440250031402700277708
           G,2537317702206760000048,1537237525733067306208,3155305026452342164008
```

```
     4,13400642034501500055B,0062016704730675137 7B,1677042204560420256 0B
     1,2522655600000000000B,1037167713371340100 B,5640000000000000000B
     J,0037247702370240000 B,0440021427772537317 7B/
0000011       DATA KPOINT/
     1 1223274025007140000B,0000064002001477264 B,2400704000000000000B
     2,0000044002000277003 B,0377154027773277337 7B,3040260072400000000B
     3,0000044002000277003 B,0377264026772437777 7B,0037007400362776277 7B
     4,2774662206560621210 0B,2122215600030040000 1B,2741270367400000000B
     5,1037237525733067316 2B,3155305026452342164 0B,1340064203450150005 5B
     6,0062016704730675137 7B,5677000000000000000 0B,0000044002000277003 7B
     7,3177273727402500714 0B,0037217724762674277 1B,2765266224602157025 7B
     8,0237024000004540000 0B,2732267700771760000 0B,2640074500000000000 0B
     9,0032017727773072143 7B,1440100060400000000 0B,0026007201750477077 7B
     A,1275147016752177247 7B,2775307230661430144 0B,1000604000000000000 0B
     B,1432207123672564266 1B,2656255323502046144 5B,1245064603501530056 B
     C,0061016403670671127 2B,1472073717771337194 3B,0700574000000000000 0B
     D,0037047702372640300 0B,2440040000400200247 7B,2437707700000000000 0B
     E,0027027570465075312 518,1651215302465266730 67B,1037207714371440100 0B
     F,6040000000000000000 0B,0005014006400155000 1B,0065027206761377147 7B
     G,2176257227652761265 5B,4140264067450000000 0B/
0000011       DATA ISTART/185,687,1192,1494,1895,2198,25100,28103,32109,34111,
     1 35114,37116,40119,43121,48125,53130,55132,59135,64137,67139,70143
     2,75149,77152,80156/
0000011       LETTR=LETTER+24*ICAP
0000014       SINTHE=SIN(THETA*.01745)
0000023       COSTHE=COS(THETA*.01745)
0000032       DO 120 J=1,24
0000034       JSTAR=ISTART(J)
0000036       IF(LETTER.EQ.J) GO TO 130
0000037  120  CONTINUE
0000040  130  IF(JSTAR.GT.10000) GO TO 140
0000044       JSTART=JSTAR/100
0000046       IF(LETTR.GT.24) JSTART=JSTAR-JSTART*100
0000054       GO TO 150
0000055  140  JSTART=JSTAR/1000
0000060       IF(LETTR.GT.24) JSTART=JSTAR-JSTART*1000
0000066  150  IFLAG=0
0000067  200  JPEN=2
0000070       CALL SEPAR8(IPOINT(JSTART),IX,IY,IPEN,IFLAG,LAST)
0000077       X=XYINCH(IX+1)*COSTHE-XYINCH(IY+1)*SINTHE
0000104       Y=XYINCH(IY+1)*COSTHE+XYINCH(IX+1)*SINTHE
0000111       IF(IPEN.EQ.0) JPEN=3
0000117       CALL PLOT(X*SCALE+XPOS,Y*SCALE+YPOS,JPEN)
0000127       IFLAG=1
0000130       IF(LAST.NE.1) GO TO 200
0000136       CALL PLOT(XPOS,YPOS,3)
0000147       RETURN
0000150       END
```

```
                SUBROUTINE SEPARB (ID,IX,IY,IPEN,IFLAG,LAST)
000011          DIMENSION ID(1)
000011          LAST=0
000011          IF(IFLAG.EQ.0) LOC=12
000013      100 LOCW=(LOC-12)/60 +1
000020          IDD=ISHIFT(ID(LOCW),MOD(LOC,60))
000034          IY=IDD.AND.31
000036          IPEN=IDD.AND.32
000037          IDD=ISHIFT(IDD,-6)
000042          IX=IDD.AND.31
000050          LOC=LOC+12
000051          IDD=IDD.AND.32
000053          IF(IDD.NE.0) LAST=1
000056          RETURN
000056          END
```

```
                                IDENT   ISHIFT
                                ENTRY   ISHIFT
                                VFD     36/0LISHIFT,24/2
0  1123111-6240000002   ISHIFT  BSS     1
1
2  56110                        SA1     B1
       56220                    SA2     B2
             63320              SB3     X2
                   22631        LX6     B3,X1
                                EQ      ISHIFT
3  0400000001 +                 END
4
                   45663  STORAGE USED              10 STATEMENTS       1 SYMBOLS
                          6400 ASSEMBLY           0.137 SECONDS        3 REFERENCES
```

B. BUDDHA

1. Introduction

Program BUDDHA was originally designed to process a maximum of two data tapes according to record number, to plot right and left hand axes on a linear scale, and to overlay curves.[1] The present version of BUDDHA has been expanded to process multiple tapes according to record name and/or record number and to plot right and left hand axes on a log scale.

2. Input Revisions

There are three additional data card types and card type B[2] has three new variables defined.

Type C is the tape card, type D is the record card, and type E is the log card. A data set consists of a tape card followed by as many record cards as specified on the tape card. When all the record cards have been read, another tape card is read and the cycle continues until a blank tape card is read. An axis card is then read which is followed by a log card and as many curve cards as specified on the axis card. When all the curve cards are read another axis card and a log card are read and the cycle continues until two blank cards are read which indicate all input data has been processed.

---

1. Jones, D. L. and D. H. Stump, ELECTRA, ORESTES, AND SUPPORTING GRAPHIC DISPLAY CODES, U.S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Virginia, February 1971, pp. 169-195.

2. Ibid, p. 169.

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENT	PROGRAM NAME	TYPE	DESCRIPTION
B	1-20				Same as report description
	21-30		KARD(NS,3)		Data tape number (1, 2,3, etc.)
	31-50				Same as report description
	51-54	Not used			Blank
	55-60		DATE	Alpha-numeric	Name of record to be plotted
	61-65		IRUN	Integer	Run number
	66-70		IALT	Integer	Altitude Number
	71-75		IZ(NS)	Integer	Light intensity
	76-80	Not used			Blank
C	1-10		ITAPE	Integer	Tape number (1, 2, 3, etc.)
	11-20		NDATES	Integer	Number of record names to be processed for a particular tape.
	21-80	Not used			Blank

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENT	PROGRAM NAME	TYPE	DESCRIPTION
D	1-6		TDATE(I, ITAPE)	Alpha-numeric	Record name on data tape, where I runs from 1 to NDATES.
	4-10	Not used			Blank
	11-20		NRUNS	Integer	Total number of runs for a particular record name
	21-30		MAXALT	Integer	Total number of altitudes for a particular record name
	31-40		MAXVAL	Integer	Number of maximum value records
	41-80	Not used			Blank
E	1-4		ILOGO (INC)	Alpha-numeric	LOGO indicates a log axis
	5-8	Not used			Blank
	9-10		IPRT	Alpha-numeric	Print control for debugging
	11-16	Not used			Blank
	17-20		ILINE	Alpha-numeric	Grid line indicator
	21-25		DECADE (INC,1)	Floating	Number of decades for left hand axis

CARD TYPE	CARD COL.	VARIABLE NAME OR CONTENT	PROGRAM NAME	TYPE	DESCRIPTION
	26-30		DECADE (INC,2)	Floating	Number of decades for right hand axis
	31-80	Not used			Blank

3. Data Revisions

In the original program after all data cards were processed, the curve cards were ordered according to record number and the random access files were opened. The appropriate data tape was read and data was written on the corresponding random access file.[3] In the present version, a tape card and as many record cards as specified on the tape card are read. For each record card that is read, the total number of records and the starting block number for that particular record name are calculated. The total number of records and the starting block number are then used to calculate the appropriate record number that corresponds to the record name on that curve card. The data tape number on that curve card is multiplied by ten thousand and is then added to the record number. The curve cards are then ordered according to record number and the random access files are opened.

The reading of the data tape or tapes has one modification. Because multiples of ten thousand based on the tape number are added to the record number, all the records from a particular tape are processed and then a new tape is processed. Data from odd number tapes are stored on one random access file and data from even number tapes are stored on another random access file.

4. Plot Additions

Program BUDDHA has been expanded to plot ordinate axis data on a log scale. The data points are plotted on a positive log axis, a negative

---

3. Ibid., pp. 170-171

log axis, or on a split positive and negative log axis depending upon the behavior of the data. When the left hand axis array contains all positive data, the data points are scanned to determine the positive maximum and the positive minimum. If the number of cycles is defined by the user, the positive minimum is redefined as the difference between the positive maximum and the number cycles. If the number of cycles is not defined by the user, the number of cycles is set equal to the difference between the positive maximum and the positive minimum. The negative maximum, the negative minimum, and the number of cycles are determined in a similar manner when the data array contains all negative data. Both sets of variables are determined when the array contains positive and negative data. The process is identical for the right hand axis data if appropriate.

The length of the log axis is determined by the behavior of both the right and left hand axis data. The axes are 8.5 inches if the data arrays contain all positive or all negative data. If one array contains positive and negative data, the axes are split into a positive axis and a negative axis which are 4.18 inches long. The log axes are drawn using subroutine LOGPLT (see I.B.6). The time axis is located at the middle of the plot when the data is positive and negative and at the bottom when the data is either all positive or all negative. The time array is annotated with the minimum and maximum cutoff times as in the original program.[4]

4. Ibid., pp. 171-172

In order to begin plotting, the data array is scanned to determine the starting point and the end point of the positive and negative data for a particular curve. Any zeroes that are encountered are omitted. If the data oscillates positive and negative more than ten times, the variable is not plotted and message is printed. The logs of the data points are taken, converted into inches, and are plotted using subroutine LINES (see I.B.5) until all curves are drawn. Any points that are less than the defined minimum are set equal to the minimum. The ordinate axis is labeled and a title is written at the bottom of the plot. The right hand axis is then labeled if appropriate. An enclosing box is drawn around the plot and the paper is advanced until all plots are made, after which the plot file is closed.

5. Subroutine LINES

Subroutine LINES is a plot routine designed for drawing data curves and annotating plots. Given two data arrays which are to be plotted against each other, subroutine LINES performs a curve fit to the data in a manner such that the curve can be represented by a series of equally spaced dots, line segments or both. Given the X-coordinate and the Y-coordinate, subroutine LINES draws the appropriate symbol at the bottom of the plot.

6. Subroutine LOGPLT

Subroutine LOGPLT is a special plot routine designed to draw and annotate positive and negative log axes. The log axis is annotated with tic marks at the power of ten and intermediate tic marks at 2, 4, 6, and 8

between the powers of ten. For negative data the power of ten is preceded by a negative sign. The tic marks are perpendicular to the axis line and are bisected by it. A positive angle, in degrees, produces a positive log axis and a negative angle produces a negative log axis. A switch, IFR, is used to determine a right or left hand axis. If IFR is not equal to one a left hand axis is drawn and if IFR is equal to one a right hand axis is drawn.

7. CALCOMP Subroutines

There are three subroutines from the CALCOMP plotting package that are called by subroutine LOGPLT. The subroutines are NUMBER, PLOT, and SYMBOL.

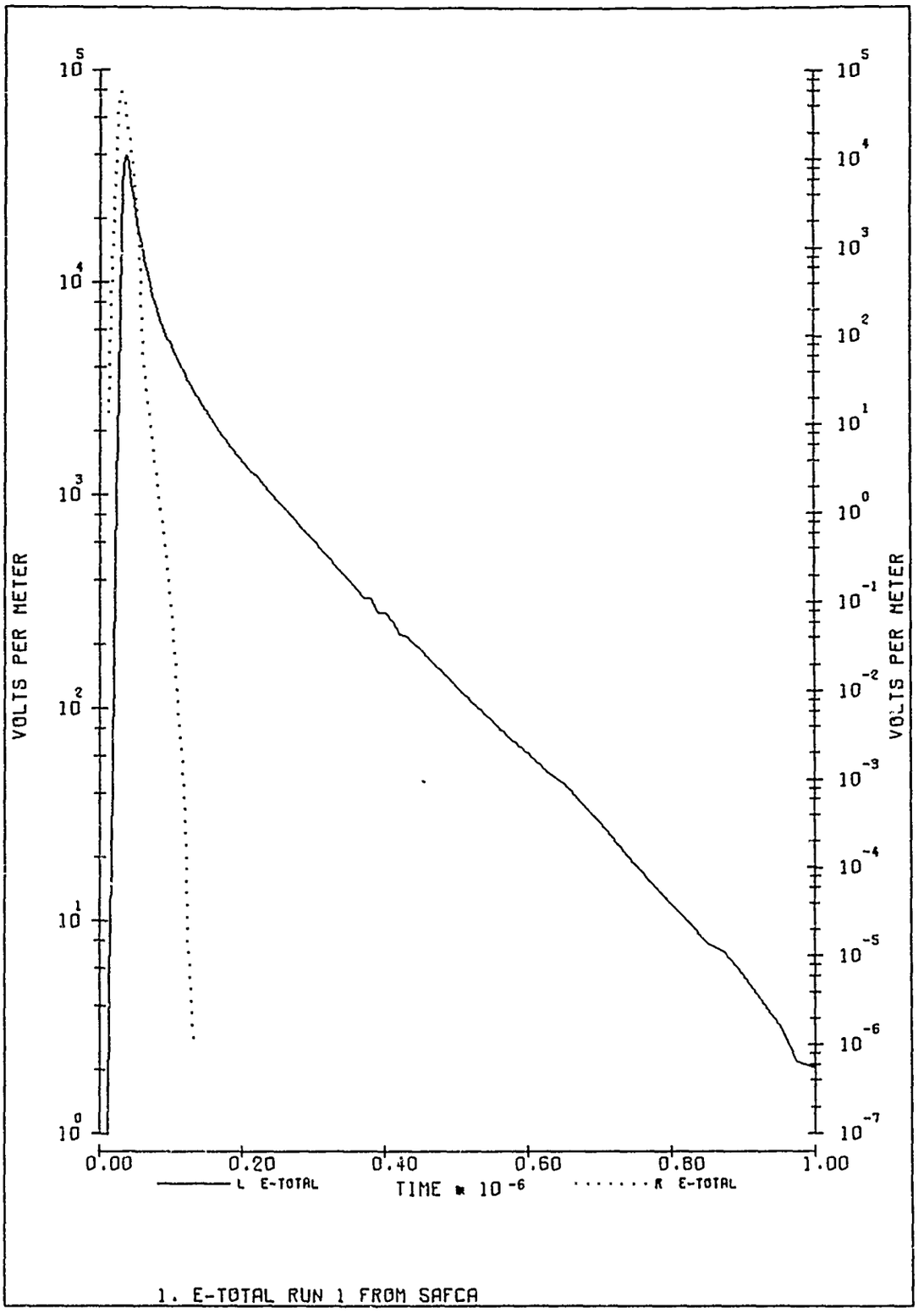
NUMBER - writes floating point numbers.

PLOT - converts pen movement from inches to actual plotter commands.

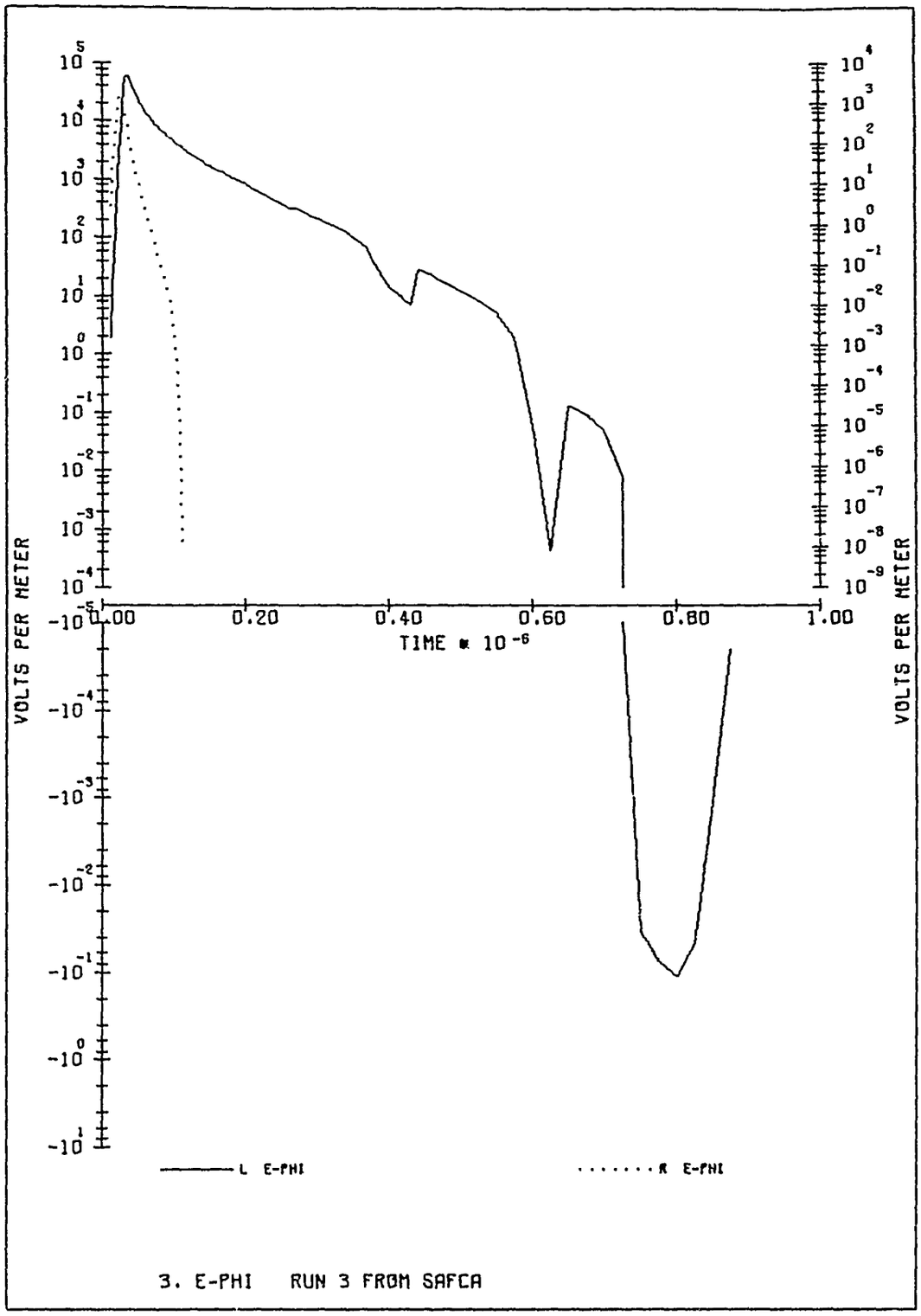
SYMBOL - writes alphanumeric text and special symbols at any angle desired.

8. Sample Output

```
 SAFCA            67         1          1         -0         -0
SAFCA1           129         -0         1         -0         -0
SAFCA2           129         -0         1         -0         -0
 B-USA           129         -0         1         -0         -0
B-USSR           129         -0         1         -0         -0
BCHINA           129         -0         1         -0         -0
SAFCA3             5         1          1         -0         -0
         2         1
         1       135
         1       264
         1       393
         1       522
         1       651
         2       780
 2-0-0-01. E-TOTAL RUN 1 FROM SAFCA                          -0.        -0.
ETO              10002         1     L          SAFCA        2          -0
ETO              10006         1     R          SAFCA        4          -0
 2-0-0-03. E-PHI   RUN 3 FROM SAFCA                          -0.        -0.
EPH              10010         1     L          SAFCA        6          -0
EPH              10014         1     R          SAFCA        8          -0
-0-0-0-0                                                     -0.        -0.
ETO              10002         1     L            1          0          0          0          0
ETO              10006         1     R            2          0          0          0          0
EPH              10010         1     L            3          0          0          0          0
EPH              10014         1     R            4          0          0          0          0
     2    0    0   51 2898  126
     6    0    0   51 2898  126
    10    0    0   51 2898  126
    14    0    0   51 2898  126
ETO              10002         1     L            1          2          1        126          1
ETO              10006         1     R            2          4          3        126          1
EPH              10010         1     L            3          6          5        126          3
EPH              10014         1     R            4          8          7        126          3
ETO              10002       126
ETO              10006       126
EPH              10010       126
EPH              10014       126
ALL DATA PROCESSED. EXIT.     NUMBER OF FRAMES PLOTTED IS   1
```



1. E-TOTAL RUN 1 FROM SAFCA



3. E-PHI RUN 3 FROM SAFCA

9. Program Listing

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

              PROGRAM RUDDHA (INPUT,OUTPUT,TAPE1,TAPE2,TAPE3=TAPE1,TAPE50,
             1 TAPE51,TAPE52)
000003        DIMENSION NC( 20,11),KARD( 50,11),IPT1(500),IPT2(500),DATA(7000),
             1 DATAR(1),DATAL(1),TIMER(2000),TIMEL(2000),IV(22),LABEL(22),
             2 IBUF(1024),IHEADR(5),TMINC( 20),TMAXC( 20),DATREC(400),
             3 TIMREC(400),IZ( 50)
000003        DIMENSION WORKT(400) ,WORKY(400), NPTS(10),ILOGO(20),DECADE(20,2),
             1 MSTART(10),NBLOCK(10,10),TDATE(10,10),NSTART(10,10)
000003        DIMENSION LUNITS(2,22)
000003        DOUBLE PRECISION LUNITX(22)
000003        INTEGER RITELL
         C
000003        DATA IV /10HETO       ,10HERA       ,10HEPH       ,10HETH       ,
             1        10HEVE       ,10HENO       ,10HEEA       ,10HEFF       ,
             2        10HCRA       ,10HCPH       ,10HCTH       ,10HMRA       ,
             3        10HMPH       ,10HMTH       ,10HCEL       ,10HSUM       ,
             4        10HSTG       ,10HBRK       ,10HAFA       ,10HEFA       ,
             5        10HELM       ,10HPRI       /
000003        DATALABEL/10HE-TOTAL  ,10HE-RADIAL ,10HE-PHI    ,10HE-THETA  ,
             1         10HE-VERT   ,10HE-NORTH  ,10HE-EAST   ,10HE-EFFECT ,
             2         10HJ-RADIAL ,10HJ-PHI    ,10HJ-THETA  ,10HJMOM-RAD ,
             3         10HJMOM-PHI ,10HJMOM-THE ,10HCOND-ELECS,10HSEC-ELECS ,
             4         10HSIGMA    ,10HATTACH   ,10HMULT-FACT ,10HENGY-FACT ,
             5         10HELEC-MOB ,10HION-RATE /
000003        DATA LUNITX/8*20H  VOLTS PER METER   ,6*20HAMPERES PER METER**2,2*
             [20H ELECTRONS/METER**3 ,20H   MHOS PER METER   ,20H      PER SECOND
             [      ,2*20H      UNITLESS      ,20HMETER**2/VOLT-SECOND,20HION PAI
             [RS/SEC-MET**3/
000003        DATA LOGO /4HLOGO/
         C
000003        COMMON //DATA
000003        EQUIVALENCE (DATA,DATAL),(DATA(2001),DATAR)
000003        EQUIVALENCE (WORKT,DATA(4001)),(WORKY,DATA(4401)),(DATREC,DATA(480
             [1))
000003        EQUIVALENCE (LUNITS,LUNITX)
         C
000003      1 FORMAT(4I2,2X,5A10,2E10.3)
000003      2 FORMAT (A10,2I10,2A10,4X,A6,3I5)
000003      3 FORMAT(1X,4I2,5A10,2E10.3)
000003      4 FORMAT(1H1)
000003      5 FORMAT(1X,A10,2I10,5X,A5,5I10,5X,A10)
000003      6 FORMAT(* ALL DATA PROCESSED, EXIT.*,5X,
             1*NUMBER OF FRAMES PLOTTED IS *,I3)
000003      7 FORMAT(1X,6I5/(1X,10E13.6))
000003      8 FORMAT(1X,10E13.6)
000003      9 FORMAT(1H )
000003     10 FORMAT(* THE RECORD # REQUESTED*,1H(,I5,1H),*IS LARGER THAN THE #
             1OF RECORDS ON TAPE UNIT*,I2,*EXIT.*)
000003     11 FORMAT (1X,A10,2I10,5X,A5,4X,A6,2I10,25X,A10)
000003     12 FORMAT (A4,4X,A2,6X,A4,2F5.0)
000003     13 FORMAT (* ORDINATE ARRAY CONTAINS ALL ZEROES FOR *,A6,I5,*,EXIT.*)
000003     14 FORMAT (* NFILES IS GREATER THAN 10 *,A10,I10)
000003     15 FORMAT (6(2X,E20.6))
000003     16 FORMAT(1X,*POSITIVE DECADES = *,F5.0,* NEGATIVES DECADES = *,F5.0)
000003     17 FORMAT(2I10)
000003     18 FORMAT (A6,4X,5I10)
000003     19 FORMAT (1X,A6,4X,5I10)
```

```
C
000003          NFRAME=1
000004          IFU=0
000005          INC=1
000006          NS=0
000007          IDX1=0
000010          IDX2=0
000011          PRINT 4
000014          NTAPES=0
000015      100 READ 17, ITAPE,NDATES
000025          IF (ITAPE.LE.0) GO TO 110
000027          NSTART(1,ITAPE)=1
C
000031          DO 105 I=1,NDATES
000033          READ 18, TDATE(I,ITAPE),NRUNS,MAXALT,MAXVAL,NGRND,NALT
000064          PRINT 19, TDATE(I,ITAPE),NRUNS,MAXALT,MAXVAL,NGRND,NALT
000076          NREC=MAXALT*MAXVAL
000100          NBLOCK(I,ITAPE)=NREC
000104          IF (I.EQ.NDATES) GO TO 105
000106          NSTART(I+1,ITAPE)=NREC*NRUNS+NSTART(I,ITAPE)
000114      105 CONTINUE
000117          PRINT 17, (NBLOCK(I,ITAPE),NSTART(I,ITAPE),I=1,NDATES)
000136          GO TO 100
C
C               READ AXIS DATA CARD
C
000137      110 READ 1,(NC(INC,I),I=1,9),TMINC(INC),TMAXC(INC)
000167          READ 12,TLOGO(INC),IPRT,ILINE,DECADE(INC,1),DECADE(INC,2)
000175          PRINT 3,(NC(INC,I),I=1,9),TMINC(INC),TMAXC(INC)
000215          NCT=NC(INC,1)
000217          IF(NCT.LE.0) GO TO 150
C
C               READ CURVE DATA CARDS
C
000220          DO 130 I=1,NCT
000222          NS=NS+1
000224          READ 2,(KARD(NS,J),J=1,4),KARD(NS,10),DATE,IRUN,IALT,IZ(NS)
000241          ITAPE=KARD(NS,3)
000243          IF(IZ(NS).LE.0) IZ(NS)=16
000247          IF (NTAPES.LT.KARD(NS,3)) NTAPES=KARD(NS,3)
000253          IF (KARD(NS,2).GT.0) GO TO 113
000256          II=1
000257      111 IF (DATE.EQ.TDATE(II,ITAPE)) GO TO 112
000274          II=II+1
000275          GO TO 111
000276      112 KARD(NS,2)=NBLOCK(II,ITAPE)*(IRUN-1)+IALT+NSTART(II,ITAPE)-1
000312      113 KARD(NS,2)=KARD(NS,2)+ITAPE*10000
000317          PRINT 11, (KARD(NS,J),J=1,4),DATE,IRUN,IALT,KARD(NS,10)
000342          KARD(NS,5)=NS
000344          KARD(NS,11) =INC
000345          DO 115 K=6,9
000347          KARD(NS,K)=0
000353      115 CONTINUE
000355          IMOD=MOD(KARD(NS,3),2)
000362          IF (IMOD.EQ.0) GO TO 120
000363          IDX1=IDX1+1
000364          GO TO 130
000365      120 IDX2=IDX2+1
```

```
000367      130 CONTINUE
000372          INC=INC+1
000373          GO TO 110
C
C               REORDER CURVE DATA ACCORDING TO RECORD NUMBER
C
000374      150 J=1
000375          JP1=2
000376          INC=INC-1
000400      160 CONTINUE
000400          DO 180 I=JP1,NS
000402          IF(KARD(J,2).LE.KARD(I,2)) GO TO 180
000405          ITEM=KARD(I,2)
000407          KARD(I,2)=KARD(J,2)
000410          KARD(J,2)=ITEM
000412          ITEM=KARD(I,5)
000413          KARD(I,5)=KARD(J,5)
000415          KARD(J,5)=ITEM
000416      180 CONTINUE
C
000421          J=J+1
000422          JP1=J+1
000423          IF(JP1.LE.NS) GO TO 160
C
000425          IDX1=2*IDX1+1
000426          IDX2=2*IDX2+1
000430          PRINT 5, ((KARD(I,J),J=1,10),I=1,NS)
000447          CALL OPENMS(51,IPT1,IDX1,0)
000462          CALL OPENMS(52,IPT2,IDX2,0)
C
C               READ DATA TAPE OR TAPES
C
000465          IT1=0
000466          IT2=0
000467          IRN=0
000470          IR1=0
000471          IR2=0
000472          IST=1
000473          ISVK1=0
000474          ISVK2=0
000475          LASTLC=0
C
000476          DO 190 IU=1,NTAPES
000477          REWIND IU
000501          READ (IU) GETSET
000506      190 CONTINUE
C
000511          DO 320 IM=1,NS
000515          LC=KARD(IM,5)
000516          ITAPE=KARD(LC,3)
000517          IREC=KARD(IM,2)-ITAPE*10000
000520          IMOD=MOD(ITAPE,2)
000522          IF (IMOD.EQ.0) GO TO 200
000524          IU=ITAPE
000525          MS=51
000526          IF (IREC.EQ.IT1) GO TO 230
000530          GO TO 210
000532      200 IU=ITAPE
```

```
000524          MS=52
000525          IF (IREC.EQ.IT2) GO TO 230
000527      210 DO 220 IA = IST,20000
000531          IF (IMOD.EQ.0) GO TO 215
000532          IT1=IT1+1
000533          IF (IREC.EQ.IT1) GO TO 225
000535          GO TO 219
000535      215 IT2=IT2+1
000537          IF (IREC.EQ.IT2) GO TO 225
000540      219 READ (IU) ID
000545      220 CONTINUE
000547          PRINT 10,IU
000555          STOP
C
000557      225 READ(IU) ID,ID,J,ID,ID,K,ID,ID,(ID,I=1,J),(DATA(I),I=1,K)
000616      230 IST=AMIN0(IT1,IT2)
000623          PRINT 7,IT1,IT2,IST,J,K,ID
000643          IF (IMOD.EQ.0) GO TO 240
000644          IF(ISVK1.NE.KARD(IM,2)) GO TO 235
000647          KARD(LC,7)=KARD(LASTLC,7)
000651          GO TO 260
000652      235 IR1=IR1+1
000654          IRN=IR1
000655          ISVK1=KARD(IM,2)
000657          GO TO 250
000657      240 IF(ISVK2.NE.KARD(IM,2)) GO TO 245
000662          KARD(LC,7)=KARD(LASTLC,7)
000664          GO TO 260
000665      245 IR2=IR2+1
000667          ISVK2=KARD(IM,2)
000671          IRN=IR2
000672      250 KARD(LC,7)=IRN
C
C
C               WRITE RANDOM ACCESS FILE
C
000674          CALL WRITMS (MS,DATA,ID,IRN)
000677      260 ISM=KARD(LC,1)
C
000701          DO 270 J=1,22
000703          JP=J
000704          IF(ISM.EQ.IV(J)) GO TO 280
000706      270 CONTINUE
C
000710      280 LOC=JP*ID+1
000714          KARD(LC,9)=JP
000715          IF (IMOD.EQ.0) GO TO 290
000716          IR1=IR1+1
000717          IRN=IR1
000720          GO TO 300
000721      290 IR2=IR2+1
000723          IRN=IR2
000724      300 KARD(LC,6)=IRN
000726          KARD(LC,8)=ID
000730          I0=LOC+ID-1
000732          CALL WRITMS (MS,DATA(LOC),ID,IRN)
000735          LASTLC=LC
000737      320 CONTINUE
```

```
000741          PRINT 9, ((KARD(I,J),J=1,10),I=1,NS)
                C
                C     BEGIN PLOTTING
                C
000741          JP=0
000742          IP=0
000743          CALL PLOTS(IBUF,1024,50)
000765          CALL PLOT (0.,-11.,-3)
000770          CALL PLOT (0.,1.25,-3)
                C
000773          DO 2000 IM=1,INC
000775          NCT=NC(IM,1)
000777          IFR=1
001000          ISYMBL=0
001001          NPVR=0
001002          NPVL=1
001003          STM= .
001004          TMAX=TMAXC(IM)
001005          TMIN=TMINC(IM)
001007          DO 325 J=1,5
001010          IHEADR(J)=NC(IM,J+4)
001015      325 CONTINUE
                C
001017          DO 400 JM=1,NCT
001020          JP=JP+1
001022          NDR=KARD(JP,6)
001024          NTR=KARD(JP,7)
001025          NPT=KARD(JP,8)
001027          MS=5
001030          IMOD=MOD(KARD(JP,3),2)
001034          IF (IMOD.EQ.0) MS=52
001036          CALL READMS(MS,DATREC,NPT,NDR)
001041          CALL READMS(MS,TIMREC,NPT,NTR)
001044          IMIN=1
001045          IMAX=NPT
001047          IMINEQ=0
001050          IMAXEQ=0
001051          IF(TMAX.EQ.0.) GO TO 350
                C
                C     EDIT DATA FOR VALUES BETWEEN TMIN AND TMAX
                C
001052          DO 342 I=1,NPT
001053          IF(TIMREC(I).LT.TMIN) GO TO 342
001056          IMIN=I
001057          IF(TIMREC(I).NE.TMIN) IMINEQ=I-1
001062          GO TO 344
001063      342 CONTINUE
001066      344 DO 346 I=IMIN,NPT
001070          IF(TIMREC(I).LT.TMAX) GO TO 346
001073          IMAX=I
001074          IF(TIMREC(I).NE.TMAX) IMAXEQ=I-1
001077          GO TO 348
001100      346 CONTINUE
001103          IMAX=NPT
001104      348 IF(IMINEQ.EQ.0) GO TO 349
001105          DATREC(IMIN)=(DATREC(IMIN)-DATREC(IMINEQ))*(TMIN-TIMREC(IMINEQ))/
                1 (TIMREC(IMIN)-TIMREC(IMINEQ)) +DATREC(IMINEQ)
001116          TIMREC(IMIN)=TMIN
```

```
001120      349 IF(IMAXEQ.EQ.0) GO TO 350
001121          DATREC(IMAX)=(DATREC(IMAX)-DATREC(IMAXEQ))*(TMAX-TIMREC(IMAXEQ))/
                1 (TIMREC(IMAX)-TIMREC(IMAXEQ)) +DATREC(IMAXEQ)
001132          TIMREC(IMAX)=TMAX
001134      350 IF(KARD(JP+4).EQ.10HL          ) GO TO 365
C
C             COLLECT DATA FOR RIGHT HAND AXIS
C
001137          DO 360 J=IMIN,IMAX
001141          NPVR=NPVR+1
001143          DATAR(NPVR)=DATREC(J)
001145          TIMER(NPVR)=TIMREC(J)
001147      360 CONTINUE
001152          TIME=TIMER(NPVR)
001153          IFR=1
001154          GO TO 380
C
C             COLLECT DATA FOR LEFT HAND AXIS
C
001155      365 DO 370 J=IMIN,IMAX
001157          NPVL=NPVL+1
001161          DATAL(NPVL)=DATREC(J)
001163          TIMEL(NPVL)=TIMREC(J)
001165      370 CONTINUE
001170          TIME=TIMEL(NPVL)
001171      380 IF(STM.LT.TIME) STM=TIME
001174          KARD(JP+8)=IMAX-IMIN+1
001200      400 CONTINUE
001202          IF (ILOGO(IM).EQ.LOGO) GO TO 700
C
C             FIND AND ADJUST MAX VALUES
C
001205          YMAGL=0.0
001206          DO 405 J=1,NPVL
001207          IF(ABS(DATAL(J)).GT.YMAGL) YMAGL=ABS(DATAL(J))
001215      405 CONTINUE
001220          CALL ADJUST(YMAGL,IPL,NL)
001222          DOL=(YMAGL*10.**IPL)/4.25
001230          IF(TMAX.EQ.0.) GO TO 407
C
001231          TMAG=TMAX
001232          CALL ADJUST(TMAG,IT,NT)
001234          DT=(TMAX-TMIN)/6.0
001237          TMAG=TMAX/(10.**IT)
001244          TM=TMIN
001245          TMINAX=TMIN/(10.**IT)
001252          GO TO 409
C
001252      407 TMAG=STM
001254          CALL ADJUST(TMAG,IT,NT)
001256          DT=(TMAG*10.**IT)/6.0
001264          TM=0.
001265          TMINAX=0.
001266      409 IF(IFR.NE.1) GO TO 415
001270          YMAGR=0.0
001271          DO 410 J=1,NPVR
001272          IF(ABS(DATAR(J)).GT.YMAGR) YMAGR=ABS(DATAR(J))
001300      410 CONTINUE
```

```
      CALL ADJUST(YMAGR,IPR,NR)
      DOR=(YMAGR*10.**IPR)/4.25
  415 IL=0
      IR=0
      LEFTLL = 0
      RITELL = 0
      IUNITL=1
      IUNITR=1
C
      DO 465 I=1,NCT
      IP=IP+1
      ISYMBL=ISYMBL+1
      ISM=KARD(IP,9)
      JLABEL=KARD(IP,10)
      IF(KARD(IP,4).EQ.10HL          ) GO TO 430
C
C        DRAW CURVES WITH RIGHT HAND AXIS SCALING
C
      IR=IR+1
      L = LEFTLL + 1
      LL = LEFTLL + KARD(IP,8)
      LEFTLL = LL
C
      ROOR=1.0/DOR
      RDT=1.0/DT
      DO 425 J=L,LL
      M = J - L + 1
      WORKT(M)=RDT*(TIMER(J)-TM)
      WORKY(M)=RDOR*DATAR(J)+4.25
  425 CONTINUE
      CALL LINES (1,2,IZ(IP),-ISYMBL)
      CALL LINES (WORKT(1),WORKY(1),0)
      CALL LINES (WORKT,WORKY,M)
      TEMPX=0.5+3.0*FLOAT(MOD(ISYMBL+1,2))
      TEMPY=-0.14*FLOAT((ISYMBL+1)/2)
      CALL LINES (TEMPX,TEMPY+0.035,0)
      TEMPX=TEMPX+0.6
      CALL LINES (TEMPX,TEMPY+0.035,1)
      CALL SYMBOL(TEMPX+0.07,TEMPY,0.07,1HR,0.0,1)
      CALL SYMBOL(TEMPX+0.21,TEMPY,0.07,LABEL(ISM),0.0,10)
      CALL SYMBOL(TEMPX+0.98,TEMPY,0.07,JLABEL,0.0,10)
      IUNITR=ISM
      GO TO 460
C
C        DRAW CURVES WITH LEFT HAND AXIS SCALING
C
  430 IL=IL+1
      L = RITELL + 1
      LL = RITELL + KARD(IP,8)
      RITELL = LL
C
      ROOR=1.0/DOL
      RDT=1.0/DT
      DO 455 J=L,LL
      M = J - L + 1
      WORKT(M)=RDT*(TIMEL(J)-TM)
      WORKY(M)=RDOR*DATAL(J)+4.25
  455 CONTINUE
```

```
      CALL LINES (1,2,IZ(IP),-ISYMBL)
      CALL LINES (WORKT(1),WORKY(1),0)
      CALL LINES (WORKT,WORKY,M)
      TEMPX=0.5+3.0*FLOAT(MOD(ISYMBL+1,2))
      TEMPY=-0.14*FLOAT((ISYMBL+1)/2)
      CALL LINES (TEMPX,TEMPY+0.035,0)
      TEMPX=TEMPX+0.6
      CALL LINES (TEMPX,TEMPY+0.035,1)
      CALL SYMBOL(TEMPX+0.07,TEMPY,0.07,1HL,0.0,1)
      CALL SYMBOL(TEMPX+0.21,TEMPY,0.07,LABEL(ISM),0.0,10)
      CALL SYMBOL(TEMPX+0.98,TEMPY,0.07,JLABEL,0.0,10)
      IUNITL=ISM
  460 CONTINUE
C
C        WRITE NEGATIVE AMPLITUDE VALUE ON LEFT HAND AXIS
C
      EXP=IPL
      Y=0.
      DY=4.25/NL
      N=2*NL+1
      IF(EXP.NE.0.) GO TO 500
      CALL NUMBER(-0.7,-0.05,.1,-YMAGL,0.0,2)
      GO TO 520
  500 CALL NUMBER(-0.7,0.05,.1,-YMAGL,0.0,2)
      IF(EXP.GT.0.0) GO TO 510
      CALL SYMBOL(-0.6,-0.15,.1,1H(,0.0,1)
      CALL NUMBER(-0.5,-0.15,.1,EXP,0.0,-1)
      CALL SYMBOL(-0.2,-0.15,.1,1H),0.0,1)
      GO TO 520
  510 CALL SYMBOL(-0.6,-0.15,.1,2H(+,0.0,2)
      CALL NUMBER(-0.4,-0.15,.1,EXP,0.0,-1)
      CALL SYMBOL(-0.2,-0.15,.1,1H),0.0,1)
  520 CALL PLOT (0.,0.,3)
C
C        DRAW LEFT HAND AXIS
C
      DO 530 I=1,N
      CALL SYMBOL(0.,Y,.1,13,90.,-2)
      CALL PLOT (0.,Y,3)
      Y=Y+DY
  530 CONTINUE
C
C        WRITE POSITIVE AMPLITUDE VALUE ON LEFT HAND AXIS
C
      CALL SYMBOL(-0.2,3.25,.1,LUNITS(1,IUNITL),90.0,20)
      IF(EXP.NE.0.) GO TO 540
      CALL NUMBER(-0.6,8.45   ,.1,YMAGL,0.0,2)
      GO TO 560
  540 CALL NUMBER(-0.6,8.55   ,.1,YMAGL,0.0,2)
      IF(EXP.GT.0.0) GO TO 550
      CALL SYMBOL(-0.6,8.35,.1,1H(,0.0,1)
      CALL NUMBER(-0.5,8.35   ,.1,EXP,0.0,-1)
      CALL SYMBOL(-0.2,8.35   ,.1,1H),0.0,1)
      GO TO 560
  550 CALL SYMBOL(-0.6,8.35   ,.1,2H(+,0.0,2)
      CALL NUMBER(-0.4,8.35   ,.1,EXP,0.0,-1)
      CALL SYMBOL(-0.2,8.35   ,.1,1H),0.0,1)
  560 CALL PLOT (0.,4.25,3)
```

```
C
C       DRAW AND LABEL TIME AXIS
C
      N=NT+1
      X=0.
      DX=6.5/NT
      EXP=IT
      TINC=(TMAG-TMINAX)/NT
      DO 570 I=1,N
      T=TINC*(I-1)+TMINAX
      CALL SYMBOL(X,4.25,.1,13,0.,-2)
      CALL NUMBER(X-.1,4.1,.1,T,0.0,2)
      CALL PLOT (X,4.25,3)
      X=X+DX
  570 CONTINUE
      CALL SYMBOL(2.5,3.9,.1,9HTIME * 10,0.0,9)
      CALL NUMBER(3.45,3.95,.07,EXP,0.0,-1)
      IF(IFR.NE.1) GO TO 660
C
C       WRITE NEGATIVE AMPLITUDE VALUE ON RIGHT HAND AXIS
C
      EXP=IPR
      Y=0.
      DY=4.25/NR
      N=2*NR+1
      IF(EXP.NE.0.) GO TO 600
      CALL NUMBER( 6.2,-0.05,.1,-YMAGR,0.0,2)
      GO TO 620
  600 CALL NUMBER( 6.2,0.05,.1,-YMAGR,0.0,2)
      IF(EXP.GT.0.) GO TO 610
      CALL SYMBOL( 6.3,-0.15,.1,1H(,0.0,1)
      CALL NUMBER( 6.4,-0.15,.1,EXP,0.0,-1)
      CALL SYMBOL( 6.7,-0.15,.1,1H),0.0,1)
      GO TO 620
  610 CALL SYMBOL( 6.3,-0.15,.1,2H(+,0.0,2)
      CALL NUMBER( 6.5,-0.15,.1,EXP,0.0,-1)
      CALL SYMBOL( 6.7,-0.15,.1,1H),0.0,1)
  620 CALL PLOT (6.,0.,3)
C
C       DRAW RIGHT HAND AXIS
C
      DO 630 I=1,N
      CALL SYMBOL(6.,Y,.1,13,90.,-2)
      CALL PLOT (6.,Y,3)
      Y=Y+DY
  630 CONTINUE
C
C       WRITE POSITIVE AMPLITUDE VALUE ON RIGHT HAND AXIS
C
      CALL SYMBOL(6.58,3.25,.1,LUNITS(1,IUNITR),90.0,20)
      IF(EXP.NE.0.) GO TO 640
      CALL NUMBER( 6.3,8.45   ,.1,YMAGR,0.0,2)
      GO TO 660
  640 CALL NUMBER( 6.3,8.55   ,.1,YMAGR,0.0,2)
      IF(EXP.GT.0.) GO TO 650
      CALL SYMBOL( 6.3,8.35,.1,1H(,0.0,1)
      CALL NUMBER( 6.4,8.35   ,.1,EXP,0.0,-1)
      CALL SYMBOL( 6.7,8.35   ,.1,1H),0.0,1)
```

```
0002165          GO TO 660
0002166      650 CALL SYMBOL( 6.3,8.35    ,.1,2H(*,0.0,2)
0002172          CALL NUMBER( 6.5,8.35    ,.1,EXP,0.0,-1)
0002176          CALL SYMBOL( 6.7,8.35    ,.1,1H),0.0,1)
0002202      660 CONTINUE
0002202          CALL SYMBOL(0.5,-1.1,.1,IHEADR,0.0,50)
0002206          GO TO 1950
C
0002207      700 IF (TMAX.EQ.0.0) GO TO 705
0002210          TMAG=TMAX
0002211          CALL ADJUST(TMAG,IT,NT)
0002214          DT=(TMAX-TMIN)/6.0
0002217          TMAG=TMAX/(10.**IT)
0002224          TM=TMIN
0002225          TMINAX=TMIN/(10.**IT)
0002232          GO TO 710
C
0002232      705 TMAG=STM
0002234          CALL ADJUST(TMAG,IT,NT)
0002236          DT=(TMAG*10.**IT)/6.0
0002244          TM=0.
0002245          TMINAX=0.
0002246      710 CONTINUE
0002246          PDEC=DECADE(IM,1)
C
C
C                DETERMINE POSITIVE MAX AND POSITIVE MIN
C
0002250          DO 800 J=1,NPVL
0002252          IF (DATAL(J).LE.0.0) GO TO 800
0002254          POSMAX=ALOG10(DATAL(J))
0002256          POSMIN=ALOG10(DATAL(J))
0002261          GO TO 850
0002261      800 CONTINUE
0002264          IPOS=0
0002265          GO TO 950
0002265      850 IPOS=1
0002266          DO 900 JJ=1,NPVL
0002270          TEMPL=DATAL(JJ)
0002272          IF (TEMPL.LE.0.0) GO TO 900
0002273          TEMPL=ALOG10(TEMPL)
0002276          IF (POSMIN.GT.TEMPL) POSMIN=TEMPL
0002301          IF (POSMAX.LT.TEMPL) POSMAX=TEMPL
0002304      900 CONTINUE
C
0002307          IF (PDEC.GT.0.0) GO TO 925
0002311          IPMAX=POSMAX
0002313          IPMIN=POSMIN
0002315          IF (IPMAX.GE.0) IPMAX=IPMAX+1
0002320          POSMAX=IPMAX
0002321          IF (IPMIN.LT.0) IPMIN=IPMIN-1
0002324          POSMIN=IPMIN
0002326          PDEC=ABS(POSMAX-POSMIN)
0002330          GO TO 950
C
0002331      925 POSMIN=POSMAX-PDEC
0002333          IPMAX=POSMAX
0002335          IPMIN=POSMIN
0002337          IF (IPMAX.GE.0) IPMAX=IPMAX+1
```

```
002342          POSMAX=IPMAX
002344          IF (IPMIN.LT.0) IPMIN=IPMIN-1
002347          POSMIN=IPMIN

C                  DETERMINE NEGATIVE MAX AND NEGATIVE MIN

002351      950 DEC=DECADE(IM,1)
002353          DO 1000 J=1,NPVL
002355          IF (DATAL(J) .GE.0.0) GO TO 1000
002357          YNEGMN=ALOG10(ABS(DATAL(J)))
002363          YNEGMX=ALOG10(ABS(DATAL(J)))
002370          GO TO 1050
002370     1000 CONTINUE
002373          INEG=0
002374          IF (IPOS.NE.0) GO TO 1150
002375          PRINT 13, KARD(IM,1),KARD(IM,2)
002404          GO TO 2000
002405     1050 INEG=1
002406          DO 1100 JJ=1,NPVL
002410          TEMPL=DATAL(JJ)
002412          IF (TEMPL.GE.0.0) GO TO 1100
002413          TEMPL=ALOG10(ABS(TEMPL))
002417          IF (YNEGMN.GT.TEMPL) YNEGMN=TEMPL
002422          IF (YNEGMX.LT.TEMPL) YNEGMX=TEMPL
002425     1100 CONTINUE
002430          IF (DEC.GT.0.0)  GO TO 1125
002432          NEGMAX=YNEGMX
002434          NEGMIN=YNEGMN
002436          IF (NEGMAX.GE.0) NEGMAX=NEGMAX+1
002441          YNEGMX=NEGMAX
002442          IF (NEGMIN.LT.0) NEGMIN=NEGMIN-1
002445          YNEGMN=NEGMIN
002447          DEC=ABS(YNEGMX-YNEGMN)
002451          GO TO 1150
C
002452     1125 YNEGMN=YNEGMX-DEC
002454          NEGMAX=YNEGMX
002456          NEGMIN=YNEGMN
002460          IF (NEGMAX.GE.0) NEGMAX=NEGMAX+1
002463          YNEGMX=NEGMAX
002465          IF (NEGMIN.LT.0) NEGMIN=NEGMIN-1
002470          YNEGMN=NEGMIN
C
002472     1150 IF (IFR.NE.1) GO TO 1290

C                  DETERMINE POSITIVE MAX AND POSITIVE MIN FOR RIGHT HAND AXIS

002474          PDECR=DECADE(IM,2)
002476          DO 1160 J=1,NPVR
002500          IF (DATAR(J).LE.0.0) GO TO 1160
002502          POSMXR=ALOG10(DATAR(J))
002504          POSMNR=ALOG10(DATAR(J))
002507          GO TO 1165
002507     1160 CONTINUE
002512          IPOSR=0
002513          GO TO 1180
002513     1165 IPOSR=1
```

```
002514          DO 1170 JJ=1,NPVR
002516          TEMPR=DATAR(JJ)
002520          IF (TEMPR.LE.0.0) GO TO 1170
002521          TEMPR=ALOG10(TEMPR)
002524          IF (POSMNR.GT.TEMPR) POSMNR=TEMPR
002527          IF (POSMXR.LT.TEMPR) POSMXR=TEMPR
002532     1170 CONTINUE
002535          IF (PDECR.GT.0.0) GO TO 1175
002537          IPMAXR=POSMXR
002541          IPMINR=POSMNR
002543          IF (IPMAXR.GE.0) IPMAXR=IPMAXR+1
002546          POSMXR=IPMAXR
002547          IF (IPMINR.LT.0) IPMINR=IPMINR-1
002552          POSMNR=IPMINR
002554          PDECR=ABS(POSMXR-POSMNR)
002556          GO TO 1180
002557     1175 POSMNR=POSMXR-PDECR
002561          IPMAXR=POSMXR
002563          IPMINR=POSMNR
002565          IF (IPMAXR.GE.0) IPMAXR=IPMAXR+1
002570          POSMXR=IPMAXR
002572          IF (IPMINR.LT.0) IPMINR=IPMINR-1
002575          POSMNR=IPMINR
002577     1180 DECR=DECADE(IM,2)
C
C            DETERMINE NEGATIVE MAX AND NEGATIVE MIN FOR RIGHT HAND AXIS
C
002601          DO 1185 J=1,NPVR
002603          IF (DATAR(J).GE.0.0) GO TO 1185
002605          YNEGN=ALOG10(ABS(DATAR(J)))
002611          YNEGX=ALOG10(ABS(DATAR(J)))
002616          GO TO 1190
002616     1185 CONTINUE
002621          INEGR=0
002622          IF (IPOSR.NE.0) GO TO 1290
002623          PRINT 13, KARD(IM,1),KARD(IM,2)
002632          GO TO 2000
002633     1190 INEGR=1
002634          DO 1200 JJ=1,NPVR
002636          TEMPR=DATAR(JJ)
002640          IF (TEMPR.GE.0.0) GO TO 1200
002641          TEMPR=ALOG10(ABS(TEMPR))
002645          IF (YNEGN.GT.TEMPR) YNEGN=TEMPR
002650          IF (YNEGX.LT.TEMPR) YNEGX=TEMPR
002653     1200 CONTINUE
002656          IF (DECR.GT.0.0) GO TO 1210
002660          NEGMXR=YNEGX
002662          NEGMNR=YNEGN
002664          IF (NEGMXR.GE.0) NEGMXR=NEGMXR+1
002667          YNEGX=NEGMXR
002670          IF (NEGMNR.LT.0) NEGMNR=NEGMNR-1
002673          YNEGN=NEGMNR
002675          DECR=ABS(YNEGX-YNEGN)
002677          GO TO 1290
002700     1210 YNEGN=YNEGX-DECR
002702          NEGMXR=YNEGX
002704          NEGMNR=YNEGN
002706          IF (NEGMXR.GE.0) NEGMXR=NEGMXR+1
```

```
002711          YNEGX=NEGMXR
002713          IF (NEGMNR.LT.0) NEGMNR=NEGMNR-1
002715          YNEGN=NEGMNR
002720    1290  CONTINUE
C
002720          IF (IPOS.NE.1) GO TO 1350
002722          IF (INEG.NE.1) GO TO 1325
C
C                  LEFT HAND AXIS HAS POSITIVE AND NEGATIVE DATA
C
002724          CALL PLOT (0.,-4.39,-3)
002726          DELTAN=DEC/4.18
002730          DELTAP=PDEC/4.18
002732          YSHIFP=.14
002733          YSHIFN=-.14
002735          CALL LOGPLT (4.18,90.,ILINE,PDEC,POSMIN,DELTAP,IPRT,0)
002744          CALL LOGPLT(4.18,-90.,ILINE,DEC,YNEGMN,DELTAN,IPRT,0)
002754          CALL PLOT (0.,0.,3)
002757          YORG=-4.39
002761          YIN=1.0
002762          IF (IFR.NE.1) GO TO 1400
002764          IF (IPOSR.NE.1) GO TO 1315
002765          IF (INEGR.NE.1) GO TO 1300
C
C                  RIGHT HAND AXIS HAS POSITIVE AND NEGATIVE DATA
C
002767          DELTPR=PDECR/4.18
002771          DELTNR=DECR/4.18
002772          CALL LOGPLT (4.18,90.,ILINE,PDECR,POSMNR,DELTPR,IPRT,IFR)
003002          CALL LOGPLT (4.18,-90.,ILINE,DECR,YNEGN,DELTNR,IPRT,IFR)
003012          CALL PLOT (0.,0.,3)
003015          GO TO 1400
C
C                  RIGHT HAND AXIS HAS ALL POSITIVE DATA
C
003016   1300   DELTPR=PDECR/4.18
003020          CALL LOGPLT (4.18,90.,ILINE,PDECR,POSMNR,DELTPR,IPRT,IFR)
003030          CALL PLOT (0.,0.,3)
003033          GO TO 1400
C
C                  RIGHT HAND AXIS HAS ALL NEGATIVE DATA
C
003034   1315   DELTNR=DECR/4.18
003036          CALL LOGPLT (4.18,-90.,ILINE,DECR,YNEGN,DELTNR,IPRT,IFR)
003046          CALL PLOT (0.,0.,3)
003051          GO TO 1400
C
C                  LEFT HAND AXIS HAS ALL POSITIVE DATA
C
003052   1325   IF (IFR.EQ.1) GO TO 1330
003054          CALL PLOT (0.,0.,3)
003057          DELTAP=PDEC/8.5
003041          YSHIFP=.14
003063          CALL LOGPLT (8.5,90.,ILINE,PDEC,POSMIN,DELTAP,IPRT,0)
003072          CALL PLOT (0.,0.,3)
003075          YIN=0.0
003076          YORG=-0.15
003100          GO TO 1400
```

```
C
 1330 IF (IPOSR.NE.1) GO TO 1335
      IF (INEGR.NE.1) GO TO 1340
C
C        RIGHT HAND AXIS HAS POSITIVE AND NEGATIVE DATA
C
      CALL PLOT (0.,4.39,-3)
      DELTAP=PDEC/4.18
      DELTPR=PDECR/4.18
      DELTNR= DECR/4.18
      YSHIFP=.14
      YSHIFN=-.14
      CALL LOGPLT (4.18,90.,ILINE,PDEC,POSMIN,DELTAP,IPRT,0)
      CALL LOGPLT (4.18,90.,ILINE,PDECR,POSMNR,DELTPR,IPRT,1)
      CALL LOGPLT (4.18,-90.,ILINE,DECR,YNEGN,DELTNR,IPRT,1)
      CALL PLOT (0.,0.,3)
      YORG=-4.39
      YIN=1.0
      GO TO 1400
C
C        RIGHT HAND AXIS HAS ALL NEGATIVE DATA
C
 1335 CALL PLOT (0.,4.39,-3)
      DELTAP=PDEC/4.18
      DELTNR=DECR/4.18
      YSHIFP=.14
      YSHIFN=-.14
      CALL LOGPLT (4.18,90.,ILINE,PDEC,POSMIN,DELTAP,IPRT,0)
      CALL LOGPLT (4.18,-90.,ILINE,DECR,YNEGN,DELTNR,IPRT,1)
      CALL PLOT (0.,0.,3)
      YORG=-4.39
      YIN=0.0
      GO TO 1400
C
C        RIGHT HAND AXIS HAS ALL POSITIVE DATA
C
 1340 CALL PLOT(0.0,0.0,3)
      DELTAP=PDEC/8.5
      DELTPR=PDECR/8.5
      YSHIFP=0.14
      CALL LOGPLT (8.5,90.,ILINE,PDEC,POSMIN,DELTAP,IPRT,0)
      CALL LOGPLT (8.5,90.,ILINE,PDECR,POSMNR,DELTPR,IPRT,1)
      CALL PLOT (0.,0.,3)
      YORG=-0.15
      YIN=0.0
      GO TO 1400
C
C        LEFT HAND AXIS HAS ALL NEGATIVE DATA
C
 1350 IF (IFR.EQ.1) GO TO 1360
      CALL PLOT (0.,8.5,-3)
      DELTAN=DEC/8.5
      CALL LOGPLT (8.5,-90.,ILINE,DEC,YNEGMN,DELTAN,IPRT,0)
      YSHIFN=0.
      YORG=-8.5
      YIN=8.5
      CALL PLOT (0.,0.,3)
      GO TO 1400
```

```
0033A0     1360 IF (IPOSR.NE.1) GO TO 1370
0033A2          IF (INEGR.NE.1) GO TO 1380
                C
                C     RIGHT HAND AXIS HAS POSITIVE AND NEGATIVE DATA
                C
0033A4          CALL PLOT (0.,4.39,-3)
0033A6          DELTAN=DEC/4.18
0033B0          DELTPR=PDECR/4.18
0033B2          DELTNR=DECR/4.18
0033B3          YSHIFP=.14
0033B5          YSHIFN=-.14
0033B6          CALL LOGPLT (4.18,-90.,ILINE,DEC,YNEGMN,DELTAN,IPRT,0)
0033C6          CALL LOGPLT (4.18,90.,ILINE,PDECR,POSMNR,DELTPR,IPRT,1)
0033D6          CALL LOGPLT (4.18,-90.,ILINE,DECR,YNEGN,DELTNR,IPRT,1)
0033E6          CALL PLOT (0.,0.,3)
0033E1          YORG=-4.39
0033E3          YIN=0.0
0033E4          GO TO 1400
                C
                C     RIGHT HAND AXIS HAS ALL NEGATIVE DATA
                C
0033E4     1370 CALL PLOT (0.,8.5,-3)
0033E7          DELTAN=DEC/8.5
0033F1          DELTNR=DECR/8.5
0033F3          YSHIFN=0.
0033F4          CALL LOGPLT (8.5,-90.,ILINE,DEC,YNEGMN,DELTAN,IPRT,0)
0033F3          CALL LOGPLT (8.5,-90.,ILINE,DECR,YNEGN,DELTNR,IPRT,1)
0033F3          CALL PLOT (0.,0.,3)
0033F6          YORG=-8.5
0033F0          YIN=-8.5
0033F1          GO TO 1400
                C
                C     RIGHT HAND AXIS HAS ALL POSITIVE DATA
                C
0033F1     1380 CALL PLOT (0.,4.39,-3)
0033F4          DELTAN=DEC/4.18
0033F6          DELTPR=PDECR/4.18
0033G0          YSHIFN=-.14
0033G1          YSHIFP=.14
0033G3          CALL LOGPLT (4.18,-90.,ILINE,DEC,YNEGMN,DELTAN,IPRT,0)
0033G2          CALL LOGPLT (4.18,90.,ILINE,PDECR,POSMNR,DELTPR,IPRT,1)
0033G2          YORG=-4.39
0033G4          YIN=0.0
                C
                C     DRAW AND LABEL TIME AXIS
0033G5     1400 N=NT+1
0033G7          X=0.
0033H0          DX=6.5/NT
0033H2          EXP=IT
0033H3          TINC=(TMAG-TMINAX)/NT
0033H7          DO 1410 I=1,N
0033J0          T=TINC*(I-1)+TMINAX
0033J5          CALL SYMBOL (X,YIN,.1,13,0.,-2)
0033J7          CALL NUMBER (X-.1,YIN-.15,.1,T,0.0,2)
0033K1          CALL PLOT (X,YIN,3)
0033K4          X=X+DX
0033K6     1410 CONTINUE
0033L1          CALL SYMBOL (2.5,YIN-.35,.1,9HTIME * 10.0.0.9)
```

```
000517          CALL NUMBER (3.45,YIN-.30,.07,EXP,0.,-1)
000525          IL=0
000526          IP=0
000527          LEFTLL=0
000530          RITELL=0
000531          IUNITL=1
000532          IUNITR=1
000533          NFILES=0
                C
000534          DO 1900 I=1,NCT
000535          IP=IP+1
000537          ISYMBL=ISYMBL+1
000540          ISM=KARD(IP,9)
000542          JLABEL=KARD(IP,10)
000543          PRINT 5, KARD(IP,1),KARD(IP,2),KARD(IP,8)
000555          IF (KARD(IP,4).EQ.10HP          ) GO TO 1750
                CCCC
                CCCC       DRAW LOG PLOTS WITH LEFT HAND AXIS
                CCCC
000560          IL=IL+1
000561          L=RITELL+1
000563          LL=RITELL+KARD(IP,8)
000565          RITELL=LL
000566          ISTART=L
000567          NFILE=1
000570    1420  ICOUNT=0
000571          MCOUNT=0
000572          DO 1425 LPTS=ISTART,LL
000574          IF (DATAL(LPTS).LT.0.0) GO TO 1450
000576          IF (DATAL(LPTS).NE.0.0) GO TO 1430
000577          MCOUNT=MCOUNT+1
000600    1425  CONTINUE
000603    1430  ISTART=ISTART+MCOUNT
000605          MCOUNT=0
000605          DO 1440 LPTS=ISTART,LL
000607          IF (DATAL(LPTS).GT.0.0) GO TO 1435
000612          IF (DATAL(LPTS).LT.0.0) GO TO 1450
000613          MCOUNT=MCOUNT+1
000615          GO TO 1440
000615    1435  ICOUNT=ICOUNT+1
000617    1440  CONTINUE
000622          NPTS(NFILE)=ICOUNT
000624          MSTART(NFILE)=ISTART
000625          GO TO 1550
000626    1450  IF (ICOUNT.LE.0) GO TO 1475
000630          NPTS(NFILE)=ICOUNT
000631          MSTART(NFILE)=ISTART
000633          NFILE=NFILE+1
000634          IF (NFILE.LE. 10) GO TO 1475
000636          NFILES=NFILES + 1
000637          PRINT 14 , KARD (IP,1), KARD(IP,2)
000647          IF (NFILES.GE. NCT) GO TO 2000
000652          GO TO 1900
000652    1475  ICOUNT=0
000653          ISTART=ISTART+MCOUNT+ICOUNT
000656          IF (ISTART.GT.LL) GO TO 1550
000661          MCOUNT=0
000661          DO 1520 LPTS=ISTART,LL
```

```
003662          IF (DATAL(LPTS).LT.0.0) GO TO 1480
003664          IF (DATAL(LPTS).GT.0.0) GO TO 1525
003666          MCOUNT=MCOUNT+1
003667          GO TO 1500
003670    1480  JCOUNT=JCOUNT+1
003672    1500  CONTINUE
003675    1525  NPTS(NFILE) =JCOUNT
003677          MSTART(NFILE)=ISTART
003701          ISTART=ISTART+MCOUNT+JCOUNT
003703          IF (ISTART.GT.LL) GO TO 1550
003706          NFILE=NFILE+1
003707          IF (NFILE.LE.10) GO TO 1420
003711          NFILES=NFILES + 1
003712          PRINT 14 , KARD (IP,1), KARD(IP,2)
003722          IF (NFILES.GE. NCT) GO TO 2000
003725          GO TO 1900
003725    1550  CONTINUE
C
003725          CALL LINES (1.2,IZ(IP),-ISYMBL)
003732          DO 1700 II=1,NFILE
003734          K=MSTART(II)
003736          J=NPTS(II)+K-1
003740          MFIRST=0
003741    1560  IF (DATAL(K).LT.0.0) GO TO 1600
003743          IF (DATAL(K).GT.0.0) GO TO 1575
003746          K=K+1
003747          IF (K.GT.J) GO TO 1700
003752          GO TO 1560
003752    1575  YMIN=POSMIN
003754          YP2=DELTAP
003755          YSHIFT=YSHIFP
003757          GO TO 1625
003757    1600  YMIN=YNEGMN
003761          YP2=-DELTAN
003762          YSHIFT=YSHIFN
003764    1625  M=0
003765          DO 1675 IK=K,J
003767          TEMPL=DATAL(IK)
003771          IF (TEMPL.EQ.0.0) GO TO 1675
003772          M=M+1
003773          IF (MFIRST.EQ.0) MFIRST=M
003775          TEMPL=ALOG10(ABS(TEMPL))
004001          IF (TEMPL.LT.YMIN) TEMPL=YMIN
004004          WORKT(M)=(TIMEL(IK)-T4)/DT
004010          WORKY(M)=(TEMPL-YMIN)/YP2+YSHIFT
004014    1675  CONTINUE
004017          IF (II.NE.1) GO TO 1690
004021          CALL LINES (WORKT (MFIRST),WORKY(MFIRST),0)
004024    1690  CALL LINES (WORKT,WORKY,M)
004027          IF (II.EQ.NFILE) GO TO 1700
004031          CALL LINES (WORKT(M),YSHIFT,1)
004034          CALL LINES (WORKT(M),-YSHIFT,0)
004041    1700  CONTINUE
004044          TEMPX=0.5+3.5*FLOAT(MOD(ISYMBL+1,2))
004054          TEMPY=-0.14*FLOAT((ISYMBL+1)/2)+YORG
004061          CALL LINES (TEMPX,TEMPY+0.035,0)
004065          CALL LINES (TEMPX+.60,TEMPY+0.035,1)
004074          CALL SYMBOL(TEMPX+.67,TEMPY+0.07,1HL,0.0,1)
```

```
004102          CALL SYMBOL(TEMPX+.87,TEMPY,0.07,LABEL(ISM),0.0,10)
004111          CALL SYMBOL(TEMPX+1.58,TEMPY,0.07,JLABEL,0.0,10)
004117          IUNITL=ISM
004121          GO TO 1900
004121    1750  IR=IR+1
004123          L=LEFTLL +1
004124          LL=LEFTLL +KARD(IP+8)
004126          LEFTLL=LL
004127          ISTART=L
004131          NFILE=1
004132    1760  ICOUNT=0
004133          MCOUNT=0
004134          DO 1770 LPTS=ISTART,LL
004136          IF (DATAR(LPTS).LT.0.0) GO TO 1810
004140          IF (DATAR(LPTS).NE.0.0) GO TO 1780
004141          MCOUNT=MCOUNT+1
004142    1770  CONTINUE
004145    1780  ISTART=ISTART+MCOUNT
004147          MCOUNT=0
004147          DO 1800 LPTS=ISTART,LL
004151          IF (DATAR(LPTS).GT.0.0) GO TO 1790
004154          IF (DATAR(LPTS) .LT. 0.0) GO TO 1810
004155          MCOUNT=MCOUNT+1
004157          GO TO 1800
004157    1790  ICOUNT=ICOUNT+1
004161    1800  CONTINUE
004164          NPTS(NFILE)=ICOUNT
004166          MSTART(NFILE)=ISTART
004167          GO TO 1860
004170    1810  IF (ICOUNT.LE.0) GO TO 1820
004172          NPTS(NFILE)=ICOUNT
004173          MSTART(NFILE)=ISTART
004175          NFILE=NFILE+1
004176          IF (NFILE.LE.10) GO TO 1820
004200          NFILES=NFILES+1
004201          IF (NFILES.GE.NCT) GO TO 2000
004203          GO TO 1900
004204    1820  JCOUNT=0
004205          ISTART=ISTART+MCOUNT+ICOUNT
004210          IF (ISTART.GT.LL) GO TO 1860
004213          MCOUNT=0
004213          DO 184- LPTS=ISTART,LL
004214          IF (DATAR(LPTS).LT.0.0) GO TO 1830
004216          IF (DATAR(LPTS).GT.0.0) GO TO 1850
004220          MCOUNT=MCOUNT+1
004221          GO TO 184-
004222    1830  JCOUNT=JCOUNT+1
004224    1840  CONTINUE
004227    1850  NPTS(NFILE)=JCOUNT
004231          MSTART(NFILE)=ISTART
004233          ISTART=ISTART+MCOUNT+JCOUNT
004235          IF (ISTART.GT.LL)GO TO 1860
004240          NFILE=NFILE+1
004241          IF (NFILE.LE.10) GO TO 1760
004243          NFILES=NFILES+1
004244          IF (NFILES.GE.NCT) GO TO 2000
004246          GO TO 1900
004247    1860  CONTINUE
```

```
004247          CALL LINES (1,2,IZ(IP),-ISYMBL)
004254          DO 1895 II=1,NFILE
004256          K=MSTART(II)
004260          J=NPTS(II)+K-1
004262          MFIRST=0
004263    1870  IF (DATAR(K).LT.0.0) GO TO 1880
004265          IF (DATAR(K).GT.0.0) GO TO 1875
004270          K=K+1
004271          IF (K.GT.J) GO TO 1895
004274          GO TO 1870
004274    1875  YMIN=POSMNR
004276          YP2=DELTPR
004277          YSHIFT=YSHIFP
004301          GO TO 1885
004301    1880  YMIN=YNEGN
004303          YP2=-DELTNR
004304          YSHIFT=YSHIFN
004306    1885  M=0
004307          DO 1890 IK=K,J
004311          TEMPR=DATAR(IK)
004313          IF (TEMPR.EQ.0.0) GO TO 1890
004314          M=M+1
004315          IF (MFIRST.EQ.0) MFIRST=M
004317          TEMPR=ALOG10(ABS(TEMPR))
004323          IF (TEMPR.LT.YMIN) TEMPR=YMIN
004326          WORKT(M)=(TIMER(IK)-TM)/DT
004332          WORKY(M)=(TEMPR-YMIN)/YP2+YSHIFT
004336    1890  CONTINUE
004341          IF (II.NE.1) GO TO 1892
004343          CALL LINES (WORKT(MFIRST),WORKY(MFIRST),0)
004346    1892  CALL LINES (WORKT,WORKY,M)
004351          IF (II.EQ.NFILE) GO TO 1895
004353          CALL LINES (WORKT(M),YSHIFT,1)
004356          CALL LINES (WORKT(M),-YSHIFT,0)
004363    1895  CONTINUE
004366          TEMPX=0.5+3.5*FLOAT(MOD(ISYMBL+1,2))
004376          TEMPY=-0.14*FLOAT((ISYMBL+1)/2)+YORG
004403          CALL LINES (TEMPX,TEMPY+0.035,0)
004407          CALL LINES (TEMPX+.60,TEMPY+0.035,1)
004416          CALL SYMBOL(TEMPX+.67,TEMPY,0.07,1HR,0.0,1)
004424          CALL SYMBOL(TEMPX+.87,TEMPY,0.07,LABEL(ISM),0.0,15)
004433          CALL SYMBOL(TEMPX+1.58,TEMPY,0.07,JLABEL,0.0,10)
004441          IUNITR=ISM
004443    1900  CONTINUE
004446          CALL PLOT (0.,YORG,-3)
004450          CALL SYMBOL (-.63,3.25,.1,LUNITS(1,IUNITL),90.,20)
004455          CALL SYMBOL (0.5,-1.05,.1,IHEADR,0.0,50)
004461          IF (IFR.NE.1) GO TO 1950
004463          CALL SYMBOL (6.73,3.25,.1,LUNITS(1,IUNITR),90.,20)
C
C
C               DRAW ENCLOSING BOX AND THEN ADVANCE PAPER
C
004470    1950  CALL PLOT (6.8,-1.25,3)
004473          CALL PLOT (-0.8,-1.2 ,2)
004476          CALL PLOT (-0.8,9.15,2)
004501          CALL PLOT (6.8,9.15,2)
004504          CALL PLOT (6.8,-1.25,2)
```

```
004507          CALL PLOT (11.,  0., -3)
004512     2000 CONTINUE
004515          CALL PLOT (11.,  0.,999)
004517          PRINT 6, NFRAME
004525          STOP
004527          END
```

```
                SUBROUTINE LINES (X,Y,KZ)
000006          DIMENSION X(1),Y(1),ZEND(5),JK(5)
000006          DIMENSION NJJ(10),XZ(10),XZEND(25),NJK(25),LOC(10)
000006          DATA                    NJJ/1,1,2,1,2,3,4,3,4,4/,LOC/0,1,2,4,5,7,10,
                [14,17,21/
000006          DATA XZ/1000.,0.08,0.1,0.040,0.15,0.15,0.2,0.2,0.25,0.25/,NJK/1,3,
                [1,2,3,1,2,1,2,3,1,2,3,4,1,2,3,1,2,3,4,1,2,1,2/,XZEND/100.,0.08,0.0
                [5,0.1,0.040,0.1,0.15,0.05,0.1,0.15,0.05,0.1,0.15,0.2,0.1,0.15,0.2,
                [0.1,0.15,0.2,0.25,0.1,0.15,0.2,0.25/
000006          IF (KZ.GT.0) GO TO 105
000010          IF (KZ.EQ.0) GO TO 40
000010          XINT=Y(1)
000011          SCALE=X(1)
000012          II=MOD(-KZ,10)
000017          RETURN
000017    40    IF (II.GT.1) GO TO 50
000023          CALL PLOT (X(1),Y(1),3)
000025          RETURN
000026       50 CONTINUE
000026          JJ=NJJ(II)
000030          Z=XZ(II)*SCALE
000032          DO 100 I=1,JJ
000036          IJ=I+LOC(II)
000040          ZEND(I)=XZEND(IJ)*SCALE
000043          IF(NJK(IJ).NE.1) GO TO 60
000045          ASSIGN 140 TO J
000046          GO TO 90
000046    60    IF(NJK(IJ).NE.2) GO TO 70
000051          ASSIGN 130 TO J
000052          GO TO 90
000052    70    IF(NJK(IJ).NE.3) GO TO 80
000055          ASSIGN 150 TO J
000056          GO TO 90
000056    80    ASSIGN 160 TO J
000057    90    JK(I)=J
000061   100    CONTINUE
000064          IJ=1
000064          ZLEFT=Z
000066          JDOT1=0
000066          JDOT2=0
000067          OLDX=X(1)
000070          OLDY=Y(1)
000072          RETURN
000072      105 CONTINUE
000072          NPTS=KZ
000073          IF (II.LE.1) GO TO 210
000076          DO 200 I=1,NPTS
000077          XB=OLDX
000100          YB=OLDY
000102          OLDX=X(I)
000103          OLDY=Y(I)
000105          S=SQRT((X(I)-XB)**2+(Y(I)-YB)**2)
000116          IF (S.EQ.0.0) GO TO 200
000120          XFACT=(X(I)-XB)/S
000123          YFACT=(Y(I)-YB)/S
000126   110    CONTINUE
000126          XA=XB
```

```
000127          YA=YB
000131   115    SX=AMIN1(S,-Z+ZLEFT+ZEND(IJ))
000137          IF(SX.GT.0.0) GO TO 120
000142          IJ=IJ+1
000143          IF(IJ.LE.JJ) GO TO 110
000145          IJ=1
000146          ZLEFT=Z
000147          JDOT1=0
000150          JDOT2=0
000151          GO TO 115
000152   120    CONTINUE
000152          XB=XA+SX*XFACT
000155          YB=YA+SX*YFACT
000160          JJJ=JK(IJ)
000162          GO TO JJJ,(130,140,150,160)
000165   130    CONTINUE
000165          S=S-SX
000167          ZLEFT=ZLEFT-SX
000170          IF(S.LE.0.0) GO TO 200
000172          GO TO 110
000172   140    CONTINUE
000172          CALL PLOT (XA,YA,3)
000175          CALL PLOT (XB,YB,2)
000200          GO TO 130
000203   150    CONTINUE
000203          IF(JDOT1.NE.0) GO TO 130
000204          CALL PLOT (XA,YA,3)
000207          CALL PLOT (XA,YA,2)
000212          JDOT1=1
000213          GO TO 130
000216   160    CONTINUE
000216          IF(JDOT2.NE.0) GO TO 130
000217          CALL PLOT (XA,YA,3)
000222          CALL PLOT (XA,YA,2)
000225          JDOT2=1
000226          GO TO 130
000231   200    CONTINUE
000234          RETURN
000234   210    CONTINUE
000234          DO 220 I=1,NPTS
000236          CALL PLOT (X(I),Y(I),2)
000242   220    CONTINUE
000247          RETURN
000247          END
```

```
                SUBROUTINE ADJUST(X,I,NRET)
000006          DIMENSION Y(10),N(10)
000006          DATA Y/1.0,1.2,1.5,2. ,2.5,3.0,4.0,5.0,6.5,8.0/
000006          DATA N/5,6,5,5,5,6,4,5,5,4/
000006          IF(X.NE.0.0) GO TO 5
000007          X=1.0
000010          I=1
000010          NRET=6
000011          RETURN
000012        5 CONTINUE
000012          A=X
000013          F=10.0
000014          INC=-1
000015          I=0
000016          IF(X.LE.1.0) GO TO 10
000021          F=1.0/F
000022          INC=1
000023       10 IF(A.GT.0.8.AND.A.LE.8.0) GO TO 20
000035          I=I+INC
000036          A=A*F
000040          GO TO 10
000040       20 DO 30 J=1,9
000042          IF(A.LE.Y(J)) GO TO 40
000045       30 CONTINUE
000046          J=10
000047       40 X=Y(J)
000051          NRET=N(J)
000052          RETURN
000052          END
```

```
      SUBROUTINE LOGPLT (SOR,THETA,ILINE,DEC,XLEAST,DELTA,IPR,IFR)
C
000013      DATA LINES,I2HPR /4HLINE,2HPR/
C
000013    1 FORMAT(1X,6HLOGPLT,6E20.8)
000013    2 FORMAT (5X,3F10.0)
C
000013      ANGLE=0.
000013      XMOST=XLEAST+DEC
000015      XNCYS=ABS(XMOST-XLEAST)
000017      NCYS=XNCYS+1
000022      OD=-.2
000024      TD=-.2
000024      ODM=-.2
000025      TDM=-.26
000027      XTEN=-.34
000030      XTIC=-.44
000032      IF (IFR.NE.1) GO TO 100
000035      OD=6.41
000036      TD=6.41
000037      ODM=6.41
000041      TDM=6.34
000042      XTIC=6.11
000044      XTEN=6.21
000046  100 IF (THETA.LT.0.0) GO TO 110
000050      S=SOR
000050      YSHIFT=.14
000052      GO TO 125
000053  110 S=-SOR
000054      YSHIFT=-.14
000056  125 DPCYS=S/XNCYS
000060      DO 500 K=1,NCYS
000062      XK=K
000063      FPN=XLEAST+XK-1.
000065      AFPN=ABS(FPN)
000067      IF (THETA.GT.0.0) GO TO 150
000072      YRG=(XK-1.)*DPCYS-.05
000075      CD=YRG+YSHIFT
000076      YTIC=CD+.05
000100      CALL PLOT (XTIC,YTIC,3)
000103      CALL SYMBOL (XTIC,YTIC,.1,15,0.0,-2)
000107      GO TO 175
000113  150 CONTINUE
000113      CD=(XK-1.)*DPCYS+.09
000117      YRG=CD+YSHIFT
000121  175 CONTINUE
000121      CALL NUMBER (XTEN,CD,.1,10.,0.,-1)
000125      IF (AFPN.GT.9.) GO TO 250
000134      IF (AFN.LT.0.0) GO TO 200
000135      XRG=OD
000137      GO TO 350
000137  200 CONTINUE
000137      XRG=ODM
000141      GO TO 350
000141  250 CONTINUE
000141      IF (FPN.GE.0.) GO TO 300
000143      XRG=TDM
```

```
000144          GO TO 350
000145      300 CONTINUE
000145          XRG=TD
000147      350 CONTINUE
000147          IF (IPR.NE.I2HPR) GO TO 375
000157          PRINT 1, XRG,YRG
000161      375 CONTINUE
000161          CALL NUMBER (XRG,YRG,.07,FPN,ANGLE,-1)
000165      400 CONTINUE
000165      500 CONTINUE
000173          IDEC=DEC
000174          THETA1=ABS(THETA)
000176          X=0.0
000177          IF (IPR.EQ.1)  X=6.0
000203          FY=S+YSHIFT
000205          CALL PLOT ( X,FY,3)
000216          CALL SYMBOL ( X,FY,.14,13,THETA1,-2)
000214          CALL PLOT ( X,FY,3)
000217          DO 700 I=1,IDEC
000224          XI=I
000225          TEM=(DEC-XI+1.)*DPCYS
000230          D2=TEM-.099*DPCYS
000232          D4=TEM-.398*DPCYS
000235          D6=TEM-.222*DPCYS
000240          D8=TEM-.097*DPCYS
000242          DT=ABS(TEM-1.*DPCYS)
000246          IF (THETA.LT.0.0) DT=-DT
000250          Y2=D2+YSHIFT
000252          Y4=D4+YSHIFT
000253          Y6=D6+YSHIFT
000255          Y8=D8+YSHIFT
000256          YT=DT+YSHIFT

C               DEBUG PRINT

000260          IF(IPR.NE.I2HPR) GO TO 585
000263          PRINT 1, Y2,Y4,Y6,Y8,YT,TEM
000302      585 CONTINUE
000302          CALL SYMBOL (X ,Y8,.1,13,THETA1,-2)
000306          CALL PLOT (X,Y8,3)
000311          CALL SYMBOL (X ,Y6,.1,13,THETA1,-2)
000315          CALL PLOT (X,Y6,3)
000320          CALL SYMBOL (X ,Y4,.1,13,THETA1,-2)
000324          CALL PLOT (X,Y4,3)
000327          CALL SYMBOL (X ,Y2,.1,13,THETA1,-2)
000333          CALL PLOT (X,Y2,3)
000336          CALL SYMBOL (X ,YT,.14,13,THETA1,-2)
000342          CALL PLOT (X,YT,3)
000345      700 CONTINUE
C
000353          RETURN
000354          END
```

C. GREEK

1. Introduction

Subroutine GREEK is the controlling routine for a set of three subroutines which together draw the letters of the Greek alphabet using the CALCOMP plotting system. Subroutine GREEK can draw all 24 Greek letters in either upper or lower case. The user may specify any height and any orientation angle within the limitations of the plotting equipment.

2. Calling Procedure

Subroutine GREEK is called in Fortran by the following statement:

CALL GREEK(XPOS, YPOS, SCALE, LETTER, THETA, ICAP) with the arguments defined as follows:

XPOS - the X position (in inches) of the lower left-hand corner (before rotation) of the letter.

YPOS - the y position (in inches) of the lower left-hand corner (before rotation) of the letter.

SCALE - the scale height(in inches) desired for the letter. Some lower case letters will be drawn slightly smaller to maintain proportionality between letters. A height which is a multiple of 31 times the plotter increment is recommended for best results.

LETTER - an integer number (1 to 24) specifying the position in the Greek alphabet of the letter to be drawn.

THETA - orientation angle (in degrees) at which the letter is to be drawn. 0^{o} - letter is vertical: positive angle - rotates letter counterclockwise.

ICAP - selector for upper or lower case letters. ICAP = 0 -- lower case, ICAP=1 -- upper case.

Subroutine GREEK was written with the assumption that the user will open the plot file and establish an origin prior to calling the subroutine.

3. Data Control

The data points for each letter were selected by first drawing the letter on a 31 x 31 grid and then approximating all curved lines by straight line segments. The coordinates of the starting and stopping points of each line segment were then recorded along with the pen up-down commands. Next, in order to minimize the amount of computer storage space needed for the subroutine, the data points were packed such that each 60 bit word contains the x and y coordinates and pen commands for five end points. When a letter is being drawn, the particular data points for that letter are then unpacked by subroutine SEPAR8 described below.

4. Subroutine SEPAR8

Subroutine SEPAR8 unpacks the 60 bit computer words and separates the data into x and y coordinates and pen commands. It performs this operation by using a series of left and right shifts and masking functions.

5. Function Subroutine ISHIFT

Function subroutine ISHIFT performs left or right shifts of 1 to 60 bits. It is written in COMPASS (Control Data Corporation assembly language) for use on a CDC 6000 series computer.

6. Special Considerations

Since the data points have been packed into 60 bit words and function subroutine ISHIFT is written in COMPASS, the subroutine GREEK package presented here can only be utilized on a CDC 6000 series computer. The package requires 633 (octal) words of core and utilizes the CALCOMP routine PLOT.

7. Sample Output

(height of letters = 0.31)

α β γ δ ε ζ η θ ι κ λ μ

ν ξ ο π ρ σ τ υ φ χ ψ ω

Α Β Γ Δ Ε Ζ Η Θ Ι Κ Λ Μ

Ν Ξ Ο Π Ρ Σ Τ Υ Φ Χ Ψ Ω

8. Program Listing

```
      SUBROUTINE GREEK(XPOS,YPOS,SCALE,LETTER,THETA,ICAP)
C
C                        LETTER CODES
C
C      1 ALPHA        7 ETA         13 NU           19 TAU
C      2 BETA         8 THETA       14 XI           20 UPSILON
C      3 GAMMA        9 IOTA        15 OMICRON      21 PHI
C      4 DELTA       10 KAPPA       16 PI           22 CHI
C      5 EPSILON     11 LAMBDA      17 RHO          23 PSI
C      6 ZETA        12 MU          18 SIGMA        24 OMEGA
C
C        LOWER CASE -- ICAP=0         UPPER CASE -- ICAP=1
C
000011      DIMENSION XYINCH(32),IPOINT(159),JPOINT(56),KPOINT(47),ISTART(24)
000011      EQUIVALENCE (IPOINT(57),JPOINT(1)),(IPOINT(113),KPOINT(1))
000011      DATA XYINCH/
           1 .03125,.0625,.09375,.125,.15625,.1875,.21875,.25,.28125,.3125,
           2 .34375,.375,.40625,.4375,.46875,.5,.53125,.5625,.59375,.625,
           3 .65625,.6875,.71875,.75,.78125,.8125,.84375,.875,.90625,.9375,
           4 .96875,1.0/
000011      DATA IPOINT/
           1 3125235221461642134608,1040054102440051005558,0262056510661366616648,
           2,2157234523412740305608,7142000000000000000008,0000036705731076137778,
           3,2077227623742371226608,1763136105611321166608,2057225523532350214558,
           4,1044124404464151000008,2533217115641401135208,1340145214611267107228,
           5,0573037340700000000008,2333207615771277107608,0773107020602255235188,
           6,2245204214400640024208,0050005602630565516708,2127126604640161005688,
           7,0051014604431241214008,0015575500000000000008,0037037517752076207778,
           8,1577117506720365005508,0460245204621441742208,5040000000000000000008,
           9,0051017203720470024008,0425077113721672227608,2465246262400000000008,
           A,0013006401710375061708,1277157517712064205308,1746154212400640034288,
           B,0146005300206050000608,0525034704430641104608,1240144156440000000008,
           C,0320004020260253604008,0135037705762241244008,2642122240400000000008,
           D,0432025200400212044708,0746134616472052227208,2012204721462246645038,
           E,0027036704600452136008,0440124417522600236308,6367000000000000000008,
           F,1037107512742074207508,1175057303710260036308,0061206106210356015388,
           G,0050014504442043204108,1640524000000000000008,1327166620642261235588,
           H,2352224620431641134008,1040054103430146005208,0055016103640566106788,
           I,5367000000000000000008,3127306602660165006308,0061072607402226624008,
           J,0600036404700673127508,1675217423722467246648/
000011      DATA JPOINT/
           1 2260165512550756040108,4364000000000000000008,2531237016671367106788,
           2,0566036401610055002208,0146034305411040134008,1641204322462352235528,
           3,2261206416665367000008,2627036702660165006308,0002122712421341144088,
           4,6040000000000000000008,0027036704600464036008,0350054310401440174188,
           5,2144235123602204610708,1427206622052463256008,2557245422522051145088,
           6,1152055103520154005708,0060016303650566116708,1467153750400000000038,
           7,0032017304730572174108,2040234024412333404088,0031027003660363055488,
           8,0751115013501652226008,2470257115375140000008,0527026401620056005288,
           9,0140034206401040134208,1444145714041542204088,2240254227463052305688,
           A,2762266463670000000008,0000044002001477264408,2400304006146754000008,
           B,0037217724762674277108,2766266324612160026008,2120245726552752274688,
           C,2043244121400040024708,4240000000000000000008,2632267700770237024088,
           D,0000444000000000000008,0000147730404040000008,2732277700770237024088,
           E,0000274027452220170008,1723575500000000000008,0032007725770040254088,
           F,6545000000000000000008,0370477023702400000008,0440250031402700277788,
           G,2537317702206760000008,1637237525733067316208,3155305026452342164088
```

```
        H,134006420345015000055B,0062016704730675137/B,16770422045604202560B
        I,25226556000000000000,0B,1037167713371340100OB,56400000000000000000B
        J,0037047702370240000,0B,24400214277725373177B/
000011   DATA KPOINT/
        1 1223274425007140000004,00000440020014772641B,2400704000000000000008
        2,0000044302000277005/B,0377154027773277343/B,3040200072400000000008
        3,0000044000200027700537B,0377264026772437747/B,003700740036277627378
        4,2774452200502622100532122215000030040000018,2741270367400000000008
        5,1037237525733067310235,3155305026452342164UB,1340064203450150005S8
        6,0062016707473057513775,56770000000000000000B,000004400200002770037B
        7,317727372740250671704,0037217724702674277718,2705260224502157025778
        8,0523712400000454300005,2732267700717600041B,264067450000000000000008
        9,003200177277367214570,14401000060400000000UB,002600720175047707778
        A,12751476167521772477,5,27753072306014301440B,100040400000000000000B
        B,1432271123672554200,5,2650255323552040144558,1245064603500153005668
        C,006101640367067112125,1472073717771331136,0B,070053740000000000000008
        D,00370477023720403000,244004000040020025,7/B,2437707700000000000008
        E,00270267046507531251B,1651215324652667306/B,1037207714371440100OB
        F,6040000000000000300,0B,2005014006400155004,10,0005027200761377147778
        G,2170257227052751265555,2140264067450000000008/
000011   DATA ISTART/165,687,1192,1494,1695,2198,25190,28103,32109,34111,
        1 35114,37116,40119,43121,46125,53130,55132,59135,64137,67139,70143
        2,75149,77152,80156/
000011   LETTR=LETTER+24*ICAP
000014   SINTHE=SIN(THETA*.01745)
000023   COSTHE=COS(THETA*.01745)
000032   DO 120 J=1,24
000034   JSTAR=ISTART(J)
000036   IF(LETTER.EQ.J) GO TO 130
000037 120 CONTINUE
000040 130 IF(JSTAR.GT.10000) GO TO 140
000044   JSTART=JSTAR/100
000046   IF(LETTR.GT.24) JSTART=JSTAR-JSTART*100
000054   GO TO 150
000055 140 JSTART=JSTAR/1000
000060   IF(LETTR.GT.24) JSTART=JSTAR-JSTART*1000
000066 150 IFLAG=0
000067 200 JPEN=2
000070   CALL SEPARB(IPOINT(JSTART),IX,IY,IPEN,IFLAG,LAST)
000077   X=XYINCH(IX+1)*COSTHE-XYINCH(IY+1)*SINTHE
000104   Y=XYINCH(IY+1)*COSTHE+XYINCH(IX+1)*SINTHE
000111   IF(IPEN.EQ.0) JPEN=3
000117   CALL PLOT(X*SCALE+XPOS,Y*SCALE+YPOS,JPEN)
000127   IFLAG=1
000130   IF(LAST.NE.1) GO TO 200
000136   CALL PLOT(XPOS,YPOS,3)
000137   RETURN
000140   END
```

```
                SUBROUTINE SEPARB (ID,IA,IY,IPEN,IFLAG,LAST)
000010          DIMENSION ID(1)
000011          LAST=0
000011          IF(IFLAG.EQ.0) LOC=12
000013      100 LOCW=(LOC-12)/60 +1
000020          IDD=ISHIFT(ID(LOCW),MOD(LOC,60))
000034          IY=IDD.AND.31
000036          IPEN=IDD.AND.32
000037          IDD=ISHIFT(IDD,-6)
000042          IX=IDD.AND.31
000050          LOC=LOC+12
000051          IDD=IDD.AND.32
000053          IF(IDD.NE.0) LAST=1
000055          RETURN
000056          END
```

```
                                    IDENT   ISHIFT
                                    ENTRY   ISHIFT
                                    VFD     36/0LISHIFT,24/2
0  11431.11062400003000E
1                           ISHIFT  BSS     1
2  5611,                            SA1     B1
        56220                       SA2     B2
              63320                 SB3     X6
                    22631           LX6     B3,X1
                                    EQ      ISHIFT
3  04V0000001 *                     END
4
                    45003   STORAGE USED            10 STATEMENTS       1 SYMBOLS
                            64V0 ASSEMBLY        0.143 SECONDS          3 REFERENCES
```

II. MINOR MODIFICATIONS

A. ELECTRA

1. Description of Modifications

Program ELECTRA [1,2], the ground burst EMP code, received two minor modifications since it was last documented. All input/output (I/O) operations using tape or drum were revised in order to cut running time and to make more efficient use of data tapes.

The running time was cut by replacing all lengthy I/O lists, which used implied DO loops, with a variably dimensioned subroutine (IOSAVE) which can write or read any length array as a single operation. This modification should produce up to 90% reduction in running time for the operations which it affects.

More efficient use of data tapes was accomplished by revising the contents of the data output array (OUTARR). By removing some unnecessary data items from the array and repositioning other items, the dimensions of the array were reduced from 21x7 to 21x4 resulting in a decrease of 63 words. The words in the reduced array were then packed two into one by an assembly language packing routine (PACK). The net result was to reduce

---

1. Jones, D. L. and D. H. Stump, ELECTRA, ORESTES, and Supporting Graphic Display Codes, U.S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Virginia, February 1971, pp. 1-36.

2. Barbely, J. A. and D. L. Jones, ELECTRA, An Electromagnetic Pulse Fortran Program (User's Guide), U.S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Virginia, October 1969.

the number of words written to tape, at a given time step and range step, from 147 to 43. If the number of angles used for a given run is less then 18, the number of words written to tape will be reduced even more.

2. Program Listing

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

                 PROGRAM ELECTRA(INPUT OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE7,TAPE8,
                1 TAPE5=INPUT,TAPE6=OUTPUT)
          C
000003           COMMON  E(17,201),     F(18,201),    B(18,201),     XNE(201),
                1           XNM(201),     KOUT(200),  NPRINT(200),ICOUNT(200,8),
                2        TBEGIN(200),      DMIN(16),   EFACTA(16),   EFACTB(16),
                3            HTH(17),      ECON(17),       EK(17),      SIG(17),
                4         FFACTA(17),    FFACTM(17),       TH(18),      STH(18),
                5             FK(18),     HEADER(8),     HEAD(8), OUTARR(21,4),
                6         FTEMP1(18),    FTEMP2(18),   FXTRAP(16),    THETA(18),
                7         HTHETA(17)
          C
000003           COMMON/SOURCE/        XJ1,           XJ2,           XJ3,          XJ4,
                1                      T0,            T1,            T2,           T3,
                2                      XJHUN,         CLAMDA
          C
000003           COMMON /PARAMS/              C,       COULOM,         CMU,         EPS,
                1            EPSCHO,       SOND,          RHO,      ATTACH,       XELEN,
                2            XIONOB,     XELAGE,       XIOCUM,         ALF
          C
000003           COMMON /INTEGER/       KDATA,       ISTORE,       NTIME,        LOAD,
                1             NLOAD,       LMAX,       LMAXP1,        KDIM,      KDIMP1,
                2            IRENUM,      NKOUT,        LDATA,      NBLOCK,       JDATA,
                3            JSTORE,     NSHIFT,       KDIMM1,      LMAXP4,       NPACK,
                4             IDATA
          C
000003           COMMON /RANDT/              R,          ROL,        RPDR,       RPHUR,
                1            RPHUPC,     RSTART,           DR,         HDR,        LGM1,
                2              LGND,     LGNDP1,            K,         KM1,        KNUM,
                3            NFILES,     NSHFTT,        QCON,      EMOCON,      EMOBCM,
                4             CHUDR,     DTEPS,        XTRAP,           T,        TPDT,
                5             TPHDT,      THHDT,           DT,        AGND,        XGND,
                6          OMEB(20)
          C
000003           COMMOL /SAVE/    ES(17,20), FS(18,20), BS(18,20),  XNES(20),
                1           XNMS(20)
          C
000003           DIMENSION NSTOP(16),NSTEP(16),LAB(8)
          C
000003           DATA IDATA,JDATA,KDATA,LDATA,ISTORE,JSTORE,NSTEP,NSTOP
                1    /     1,      2,      3,      4,      7,      8,
                2  2,20,200,2000,0,0,0,0,0,0,0,0,0,0,0,0,
                3          200, 1000, 10000,100000,0,0,0,0,0,0,0,0,0,0,0,0        /  NSTOP
000003           DATA LAB/8HNKOUT   =,8HKDIM    =,8HNMIN    =,8HNMAX    =,8HNSTEPS =,
                1          8HNSTART =,8HLGND    =,8HLMAX    =/
          C
000003          1 FORMAT(I10,7F10.5/(8F10.0))
000003          2 FORMAT(8I10)
000003          3 FORMAT(16I5)
000003          5 FORMAT(///35X,6HRMIN =,E10.3,8X,4HDR =,F7.4,10X,4HDT =,E10.3)
000003          6 FORMAT(///19X,4(A8,I6,10X))
000003          7 FORMAT(///(19X,8HKOUT    =,I6,18X,I6,18X,I6,18X,I6))
000003          8 FORMAT(*0IRENUM ON TAPE =*,I2,*, IRENUM ON INPUT CARD =*,I2)
000003         24 FORMAT(/10H NBLOCK = ,I5)
000003         25 FORMAT(*0ERROR...NUMBER OF OUTPUT RANGES = *,I5)
000003         26 FORMAT(2X,*RMIN READ IN IS TOO SMALL, RCAL = *,F8.3)
000003        610 FORMAT(1H1,8A10)
```

```
000003     611 FORMAT(*0*** OUTPUT DATA FILES ***)
000003     612 FORMAT(*0FILE*,I3,6X,*RANGE NO*,5X,*RANGE*,5X,*NO OF RECS*)
000003     613 FORMAT(1H ,I21,F10.4,113)
000003     614 FORMAT(*0NEXT VALUE OF I IS*,E16.8)
000003     615 FORMAT(*0ELECTRA***NORMAL TERMINATION*)
000003     999 FORMAT(8F10.0)
000003    1000 FORMAT(7A10,A8,I2)
000003    1013 FORMAT(1H1,8A10)
000003    1017 FORMAT(*0ELECTRA***ERROR TERMINATION*)
000003    1027 FORMAT(*0HEADER FOR FILE *,I2,* ON TAPE DOES NOT CHECK*)
000003    1028 FORMAT(*0HEADER ON TAPE: *,8A10)
C
000003         IERR=0
000004         DO 100 I=1,8
000006         DO 100 J=1,200
000007         ICOUNT(J,I)=0
000013     100 CONTINUE
000017         DO 101 I=1,21
000020         DO 101 J=1,4
000021         OUTARR(I,J)=0.
000025     101 CONTINUE
000031         DO 105 I=1,200
000032         KPRINT(I)=0
000033     105 CONTINUE
000035         LOAD=0
000036         LOAD=1
000037         NTIME=0
000040         NBLOCK=0
000041         NFILES=1
000042         NSHFTT=0
000043         RESTRT=1.
000044         RE=IND ISTORE
000046         PE=IND JSTORE
000050         READ 999,XJ1,XJ2,XJ3,XJ4,TC,T1,T2,T3,XJRUN,CLAMDA
000100         READ 1000,HEADER,IRENUM
000110         READ(5,999) DR,DT,TS,SEC,RMIN
000126         READ(5,1) ITHETA,(THETA(I),I=1,ITHETA)
000143         READ(5,1) IMTHET,(MTHETA(I),I=1,IMTHET)
000160         READ(5,2)NSEG,KDIM,NMAX,NSTEPS,NSTART,LGND,LMAX
000202         READ(5,3) NKPRNT,(KPRINT(I),I=1,NKPRNT)
000217         IF(NSTART.LE.0) NSTART=2
000222         IF(NSTEPS.LE.0) NSTEPS=4
000225         IF(SEC.LE.0.0) SEC=1000.
000230         READ(5,2) NKOUT
000236         IF(NKOUT.LE.0) GO TO 121
000240         READ(5,999) (ORNG(I),I=1,NKOUT)
000252         GO TO 140
000253     121 M=1
000254         READ(5,1) NDROUT,ORNG(1)
000264         TEMPOR=ORNG(1)
C
000266         DO 130 JJ=1,NDROUT
000267         READ(5,1) NROUT,DROUT
000276         DO 125 I=1,NROUT
000300         K=I+M
000302         TEMPOR=TEMPOR+DROUT
000304         ORNG(K)=TEMPOR
000306     125 CONTINUE
```

```
000310          M=M+NKOUT
000311      130 CONTINUE
000313          IF(K.LE.200) GO TO 135
000316          PRINT 25, K
000323          STOP
000325      135 NKOUT=K
000327      140 LGNDP1=LGND+1
000331          LGM1=LGND-1
000332          LMAXP1=LMAX+1
000334          LMAXP4=LMAX+4
000336          NPACK=LMAXP4**2 -1
000340          KDIMM1=KDIM-1
000341          KDIMP1=KDIM+1
000342          DO 1225 L=1,LMAXP1
000344          TH(L)=.01745329*THETA(L)
000346          STH(L)=SIN(TH(L))
000352          IF (L.EQ.LMAXP1) GO TO 1225
000354          HTH(L)=.01745329*HTHETA(L)
000356      1225 CONTINUE
000361          XTHMP=COS(TH(LGM1))/(COS(TH(LGM1))-COS(TH(LGND)))
000375          RCAL=C*DT/((TH(LGNDP1)-TH(LGND))*SQRT(1.-(C*DT/DR)**2))+1.E-5
000411          IF(RMIN.GE.RCAL) GO TO 1227
000414          IF(RMIN.LE.0.) GO TO 1226
000415          PRINT 26, RCAL
000423          STOP
000425      1226 RMIN=RCAL
000427      1227 NMIN=RMIN/(C*DT)+1.
000434          RSTART=RMIN
000435          IF (IRENUM.LE. ) GO TO 95
000436          REWIND IDATA
000440          REWIND JDATA
000442          REWIND KDATA
000444          REWIND LDATA
                C
                C       READ THE RESTART TAPE
                C
000446          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NMIN,1,1)
000464          DO 1036 J=1,NLOAD
000466          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(2),XNM(2),
               1 NLOAD,NMIN,KDIMM1,2)
000513          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1)
               1,XNE(KDIMP1),XNM(KDIMP1),NLOAD,NMIN,1,2)
000544      1036 CONTINUE
                C
000547          REWIND ISTORE
                C
000551          READ (IDATA) NFILES,NKOUT,LMAX
000562          IF(NFILES.EQ.IRENUM) GO TO 1024
000564          WRITE(6,8) NFILES,IRENUM
000574          STOP
000576      1024 NFILES=IRENUM+1
000600          WRITE (JDATA) NFILES,NKOUT,LMAX
                C
                C       DATA TAPE SETUP LOOP
                C
000611          READ(IDATA) HEAD,IRNUMT
000620          DO 1025 J=1,4
000622          IF (HEAD(J).NE.HEADER(J)) GO TO 1026
```

```
000624    1025 CONTINUE
000626         GO TO 1029
               C
               C        THE HEADER READ IN DOES NOT MATCH HEAD ON DATA TAPE
               C        PRINT HEADER,HEAD EXIT
               C
000627    1026 PRINT 1013,HEADER
000635         PRINT 1027,I
000643         PRINT 1028,HEAD
000651         PRINT 1017
000655         STOP
000657    1029 WRITE(JDATA) HEAD,IRENUM
000666         READ(IDATA) THETA,DTHETA
000675         WRITE(JDATA) THETA,DTHETA
000704         NPACKS=NPACK*2
000706         IOPT=1
000707    1031 CALL DATAPE(OUTARR,NPACKS,IOPT)
000712         IF(IOPT) 1031,66,1031
               C
000714       65 IF (IRENUM.LT.0) GO TO 96
000716         REWIND JDATA
000720         WRITE (JDATA) NFILES,NKOUT,LMAX
000731         WRITE(JDATA) HEADER,IRENUM
000740         WRITE(JDATA) THETA,DTHETA
000747       66 IF(IRENUM.GT.0) GO TO 70
               C
               C        INITIALIZE ARRAYS TO ZERO ON FIRST PASS
               C
000752         DO 68 K=1,KDIMP1
000753         XNE(K)=0.
000754         XNM(K)=0.
000755         F(LMAXP1,K)=0.
000760         B(LMAXP1,K)=0.
000763         DO 68 L=1,LMAX
000764         E(L,K)=0.
000767         F(L,K)=0.
000771         B(L,K)=0.
000774       68 CONTINUE
001000         DO 69 L=1,LMAXP1
001002         FTEMP(L)=0.0
001003       69 CONTINUE
001005       70 DO 1235 L=1,LMAX
001007         LP1=L+1
001011         ECON(L)=CMU*(TH(LP1)-TH(L))*SIN(HTH(L))
001017         FFACTA(L)=(HTH(L)-TH(L))/(TH(LP1)-TH(L))
001025         FFACTB(L)=1.-FFACTA(L)
001030         IF (L.EQ.LMAX) GO TO 1235
001031         DHTH(L)=HTH(LP1)-HTH(L)
001034         EFACTA(L)=(TH(LP1)-HTH(L))/DHTH(L)
001040         EFACTB(L)=1.-EFACTA(L)
001042    1235 CONTINUE
001045         ECON(LGND)=.5*ECON(LGND)
001047         CDT=C*DT
001051         HDT=.5*DT
001052         HDR=.5*DR
001054         DO 1245 K=1,NKOUT
001055         KOUT(K)=(ORNG(K)+HDR-RMIN)/DR+.5
001063         R=RMIN+HDR+KOUT(K)*DR
```

```
001070          NBEGIN(K)=R/CDT
001073     1245 CONTINUE
001075          PRINT 5, RMIN,DR,DT
001106          PRINT 6,   LAB(1),NROUT,LAB(2),NDIM,LAB(3),NMIN,LAB(4),NMAX
001132          PRINT 6,   LAB(5),NSTEPS,LAB(6),NSTART,LAB(7),LGND,LAB(8),LMAX
001156          PRINT 7,   (ROUT(I),I=1,NROUT)
001171          EMUCON=1.293/RHO
001173          EMUBC=EMUCON/76000.
001175          TEMP=XELEN/4.E-6
001177          QCON=-TEMP/(COULOM*ALLRUE)
001202          DTEPS = DT/EPS
001204          DTEPSG = DT/EPSGND
001206          XGND = EXP(-SGND*DTEPSG)
001212          AGND = (1.0-XGND)/SGND
001215          CMUDR = CMU*DR
001217          DTDR = DT/DR
001220          C1=C*DT/DR
001222          C2=.5*C1*RMIN/DR+NSIGN*NDIM-.900000
C
C       TIME LOOP
C
001231          DO 410 N=NMIN,NMAX
001233          T=N*DT
001236          NTIME=N
001237          CALL SECOND(TIME)
001241          IF(TIME.GE.TS) GO TO 420
001244          IF(TIME.LT.RESTRT*SEC) GO TO 103
001247          CALL RESTART
001250          RESTRT=RESTRT+1.
001252     103  IF(LOAD.NE.0) CALL SEGMNT(1)
001255     102  TMHDT=T-HDT
001257          TPHDT=T+HDT
001261          TPDT=T+DT
001263          NSHIFT=C1*N+NSHFTT-52
001271          IF(NSHIFT.LT.0) NSHIFT=0
001273          NSHFTT=NSHFTT+NSHIFT
001275          DO 2005 L=1,LMAXP1
001276          FK(L)=F(L,1)
001300          FTEMP2(L)=FTEMP(L)
001301          FTEMP(L)=F(L,2)
001303     2005 CONTINUE
001305          IF(NSHFTT.EQ.0) GO TO 104
001306          IF(NSHIFT.EQ.0) GO TO 2010
001307          CALL SHIFTY
001310          DO 2009 L=1,LMAXP1
001312          FXTRAP(L)=3.*F(L,1)-3.*FTEMP(L)+FTEMP2(L)
001322     2009 CONTINUE
001325          GO TO 120
001325     2010 DO 2011 L=1,LMAXP1
001327          F(L,1)=FXTRAP(L)
001332     2011 CONTINUE
001334      104 RSTART=RMIN+NSHFTT*DR
001340          R=RSTART-HDR
001342          Q=QCON*XJAY(RSTART,IMHD1)
001346          XNEA=XNE(1)
001347          XNMA=XNM(1)
001351          TEMP=ATTACH+ALF*(XNEA+XNMA)
001355          ARG=DT*TEMP
```

```
001356          EXPON=EXP(-ARG)
001362          IF(ARG .GE. 1.E-7) GO TO 16
001364          FACTOR = ARG
001365          GO TO 17
001366       16 FACTOR = 1. - EXPON
001370       17 XNE(1)=EXPON*XNEA+FACTOR*Q/TEMP
001375          TEMP=XJOCOM*(XNEA+XNMA)
001377          ARG = DT*TEMP
001401          EXPON = EXP(-ARG)
001404          IF(ARG .GE. 1.E-7) GO TO 18
001406          FACTOR = ARG
001407          GO TO 20
001410       18 FACTOR = 1. - EXPON
001412       20 XNM(1)=EXPON*XNMA
001414          IF(TEMP .LE. 0.0) GO TO 21
001416          XNM(1)=XNM(1)+FACTOR*ATTACH*XNEA/TEMP
001422       21 CONTINUE
001422      120 LOADM1=LOADM-1
001424          IF(LOADM1.LT.0) LOADM1=0
C
C            RANGE LOOP
C
001426          DO 370 K=1,KDIM
001430          R=R+DR
001432          RMHDR=R-HDR
001434          RPHDR=R+HDR
001435          RPDR=R+DR
001436          RPHDRC=RPHDR/C
001440          ROC=R/C
001441          DTOR=DT/R
001443          DTORDR=DTOR/DR
001445          IF (RPHDR.GE.C*TMHDT) GO TO 360
001450          KP1=K+1
001452          Q=QCON*XJAY(RPHDR,TMHDT)
001456          XNEA=XNE(KP1)
001460          XNMA=XNM(KP1)
001461          TEMP=ATTACH+ALF*(XNEA+XNMA)
001465          ARG = DT*TEMP
001467          EXPON = EXP(-ARG)
001472          IF(ARG .GE. 1.E-7) GO TO 1111
001474          FACTOR = ARG
001475          GO TO 1112
001476    1111 FACTOR = 1. - EXPON
001500    1112 XNE(KP1)=EXPON*XNEA+FACTOR*Q/TEMP
001506          TEMP=XJOCOM*(XNEA+XNMA)
001510          ARG = DT*TEMP
001511          EXPON = EXP(-ARG)
001515          IF(ARG .GE. 1.E-7) GO TO 113
001517          FACTOR = ARG
001520          GO TO 114
001521     113 FACTOR = 1. - EXPON
001523     114 XNM(KP1)=EXPON*XNMA
001526          IF(TEMP .LE. 0.0) GO TO 1115
001527          XNM(KP1)=XNM(KP1)+FACTOR*ATTACH*XNEA/TEMP
001534    1115 CONTINUE
001534          XSEC = 0.5*(XNE(K) + XNE(KP1))
001540          XION = 0.5*(XNM(K) + XNM(KP1))
001544          XJ=XJAY(R,T)
```

```
001547          CEA=COULOM*EMOCON*ASEC
001551          CXCXPX=COULOM*XIOMOB*(2.*XION+ASEC)
001555          CXAXPX=COULOM*XIOMOB*(2.*XNM(KP1)+ANE(KP1))
001562          CEAN=COULOM*EMOCON*ANE(KP1)
C
C           THETA LOOP E
C
001565          DO 185 L=1,LMAX
001566          EK(L)=E(L,K)
001571          LP1 = L+1
001573          IF (L.GE.LGND) GO TO 150
001575          EVAL=.5*(FFACTA(L)*(F(LP1,KP1)+F(LP1,K))+FFACTB(L)*(F(L,KP1)+F(L,K
               [)))
001615          SIGMA=CEA*(EMOBFN(SQRT(EVAL**2+EK(L)**2)*EMOBCN))+CXCXPX
001627          TEMP = SIGMA*DTEPS
001631          IF (TEMP.LE.15.) GO TO 162
001633          X = 0.0
001634          A = 1.0/SIGMA
001636          GO TO 180
001636      162 X = EXP(-TEMP)
001642          A = (1.0-X)/SIGMA
001645          GO TO 180
001645      150 IF (L.GT.LGND) GO TO 170
001651          SIGMA=CEA*(EMOBFN(SQRT(.25*((EK(L-1)+EK(L))**2+(FK(L)+F(L,KP1))**2
               1))*EMOBCN))+CXCXPX
001670          TEMP = SIGMA*DTEPS
001672          IF (TEMP.LE.15.) GO TO 163
001674          XAIR = 0.0
001675          AAIR = 1.0/SIGMA
001677          GO TO 165
001677      163 XAIR = EXP(-TEMP)
001703          AAIR = (1.0-XAIR)/SIGMA
001705      165 AOA1 = AAIR+AGND
001710          X = (AAIR*XGND+AGND*AAIR)/AOA1
001714          A = AAIR*AGND/AOA1
001716          GO TO 180
001716      170 IF (L.GT.LGNDP1) GO TO 160
001722          X = AGND
001723          A = AGND
001724          XJ = 1.0
001725      180 E(L,K)=X*FK(L)+A*(-AJ*(B(LP1,K)*STH(LP1)-B(L,K)*STH(L))/R/ECON(L))
001745          IF(ABS(E(L,K)).LT.1.0E-25) E(L,K)=0.
001755      185 CONTINUE                                                    TH ELOOP
C
C
C           THETA LOOP F
C
001760          DO 235 L=1,LMAXP1
001761          FK(L)=F(L,KP1)
001765          IF (L.GT.1) GO TO 200
001770          EVAL=.5*(EK(1)+E(1,KP1))
001774          GO TO 210
001774      200 IF (L.GT.LGND) GO TO 220
002000          LM1 = L-1
002001          EVAL=.5*(EFACTA(LM1)*(EK(L)+E(L,KP1))+EFACTB(LM1)*(EK(LM1)+E(LM1,K
               [P)))
002015      210 SIG(L)=CEAN*(EMOBFN(SQRT(EVAL**2+FK(L)**2)*EMOBCN))+CXAXPX
002031          TEMP = SIG(L)*DTEPS
```

```
002033          IF (TEMP.LE.15.) GO TO 213
002035          X = 0.0
002036          A = 1.0/SIG(L)
002040          GO TO 230
002041      213 X = EXP(-TEMP)
002045          A = (1.0-X)/SIG(L)
002050          GO TO 230
002051      220 IF (L.GT.LGNOP1) GO TO 230
002055          X = XGND
002056          A = AGND
002057      230 F(L,KP1)=X*FK(L)-A*(RPDR*B(L,KP1)-R*B(L,K))/RPHDR/CMUDR
002076          IF(ABS(F(L,KP1)).LT.1.E-25) F(L,KP1)=0.
002105      235 CONTINUE                                                    TH FLOOP
C
C           THETA LOOP B
C
002110          DO 245 L=2,LMAX
002111          LM1 = L-1
002113          B(L,K)=B(L,K)+DTOR*(E(L,K)-E(LM1,K))/DHTH(LM1)-DTORDR*(RPHDR*F(L,K
               1P1)-RMHDR*F(L,K))
002140          IF(ABS(B(L,K)).LT.1.E-25) B(L,K)=0.
002146      245 CONTINUE                                                    TH BLOOP
C
C           OUTPUT EDITING SECTION
C           KNUM = RANGE NO.
C           K = INDEX OF RANGE LOOP
C           KOUT(KNUM) = INDEX OF RANGE LOOP FOR RANGE KNUM
C           N = INDEX OF TIME LOOP
C           NBEGIN(KNUM) = INDEX OF TIME LOOP FOR RANGE KNUM
C
002151          DO 25. KNUM=1,NKOUT
002152          IF(K+NSHFTT+KDIM*LORDM1.EQ.KOUT(KNUM)) GO TO 260
002157      250 CONTINUE
002162          GO TO 375
002162      260 NVAL=N-NBEGIN(KNUM)
002165          IF (NVAL.LT.NSTART) GO TO 370
002167          DO 27  NN=1,NSTEPS
002171          IF (NVAL.LE.NSTOP(NN)) GO TO 280
002174      270 CONTINUE
002176          GO TO 370
002177      280 IF (MOD(NSTOP(NN)-NVAL,NSTEP(NN)).NE.0) GO TO 370
002205          CALL PRINTER
002206      370 CONTINUE                                                    R   LOOP
002211          CALL SEGMNT (2)
002212          GO TO 120
002213      380 IF(LOAD.NE.0) CALL SEGMNT (3)
002216      410 CONTINUE                                                    T   LOOP
C
C           FOR CK RUNS IRENUM IS NEG, PRINT ONLY, NO TAPES
C
002221      420 IF(IRENUM.LT.0) GO TO 510
002223          IF(NLOAD.GT.1) GO TO 430
002226          CALL SEGMNT (4)
002227          GO TO 435
002230      430 JSTORE=ISTORE
002232      435 END FILE JDATA
C
C           WRITE LAST RESTART RECORD ON LDATA
```

```
C
002234        NBLOCK=NBLOCK+1
002236        REWIND LUATA
002240        NMIN = NMAX + 1
002242        IF(NTIME.LT.NMAX) NMIN=NTIME
002245        CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NMIN,1,3)
002263        DO 505 N=1,NLOAD
002265        CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(2),XNM(2),
             1 NLOAD,NMIN,KDIMM1,4)
002312        CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1)
             1,XNE(KDIMP1),XNM(KDIMP1),NLOAD,NMIN,1,4)
002343    505 CONTINUE
C
002346        END FILE LUATA
C
C           COMPLETE PRINTED OUTPUT
C
C
002350    510 PRINT 611,HEADER
002356        PRINT 611
002362        DO 601 I=1,NFILES
002364        PRINT 612,I
002371        DO 601 K=1,NKOUT
002373        PRINT 613, K,ONNG(K),ICOUNT(K,I)
002406    601 CONTINUE
C
002413        PRINT 24, NBLOCK
002421        IF(IKENUM.GE.0) T=NMIN*DT
002426        PRINT 614,T
002434        PRINT 615
002440        STOP
002442        END
```

```
      C
      FUNCTION EMOHFN(EOVERP)
000003      IF (EOVERP.GT.0.4572) GO TO 3
000006      IF(EOVERP.GT.0.0905) GO TO 2
000011      IF(EOVERP.GT.0.0303) GO TO 1
000014      EMOBFN= 2.092/(1.0+33.0*EOVERP)
000020      RETURN
000020    1 EMOBFN=1.809202+EOVERP*(-34.1606+EOVERP*(342.685-EOVERP*1186.94))
000027      RETURN
000027    2 EMOBFN=0.169267/SQRT(EOVERP)
000033      RETURN
000033    3 IF(EOVERP.GT.60.0) GO TO 4
000037      XLN=ALOG(EOVERP)
000041      EMOBFN=0.16212*EXP(XLN*(0.04912*XLN-0.51680))
000051      RETURN
000051    4 EMOBFN=0.033684+0.6579/EOVERP
000055      RETURN
000056      END
```

```
      FUNCTION XJAY(R,T)
000005 COMMON/SOURCE/ AJ1,      XJ2,      XJ3,      XJ4,
     1                T0,       T1,       T2,       T3,
     2               XJRUN,    CLAMDA
C
000005 COMMON /PARAMS/      C,    COULOM,     CMU,     EPS,
     1          EPSGND,   SGND,     RMU,  ATTACH,  XELEN,
     2          XIOMOB,  XELMGE,  XIOCOM,     ALF
C
000005       TAU=T-R/C
000007       IF(TAU.GT.0.) GO TO 10
000012       XJAY=0.
000012       RETURN
000013    10 IF(TAU.GE.T2) GO TO 20
0000,6       XJAY=-(1.675E5*EXP(2.E8*TAU))/(1.+1.816E-4*EXP(2.5E8*TAU))/R**2
000036       RETURN
000037    20 XJAY=-(6.349E6*EXP(-.005E8*TAU))/R**2
000047       RETURN
000050       END
```

```
      SUBROUTINE RESTART
000002      COMMON  E(17,201),     F(18,201),     B(18,201),     XNE(201),
     1          XNM(201),     XOUT(200),  KPRINT(200),ICOUNT(200,8),
     2       NBEGIN(200),     DM1M(16),   EFACTA(16),   EFACTB(16),
     3           MTH(17),     ECON(17),       EK(17),      SIG(17),
     4        FFACTA(17),   FFACTB(17),       TH(18),      STH(18),
     5            FK(18),    HEADE0(8),     HEAD(8), OUTARR(21,4),
     6         FTEMP(18),   FTEMP2(18),   EXTRAP(18),    THETA(18),
     7        HTHETA(17)
      C
000002      COMMON /INTEGER/      KDATA,     ISTORE,      NTIME,       LOAD,
     1           NLOAD,       LMAX,     LMAXP1,       KDIM,     KDIMP1,
     2          IRENUM,     NKOUT,      LDATA,     NBLOCK,      JDATA,
     3          JSTORE,    NSHIFT,     KDIMM1,     LMAXP4,      NPACK,
     4           IDATA
      C
000002      COMMON /RANDT/          R,        RDC,       RPDR,      RPHDR,
     1          RPHDRC,    KSTART,        DR,        HDR,       LGM1,
     2            LGND,    LGNDP1,         K,        KP1,       KNUM,
     3          NFILES,    NSHFTT,      QCON,    EMOCON,    EMOBCN,
     4           CMUDR,     DTEPS,     XTRAP,         T,       TPDT,
     5           TPHDT,     TMHDT,        DT,      AGND,      XGND,
     6        ORNG(200)
      C
000002      COMMON /SAVE/     ES(17,20), FS(18,20), BS(18,20),  XNES(20),
     1         XNMS(20)
      C
000002    1 FORMAT(////19H NBLOCK = ,I5)
      C
000002      NBLOCK=NBLOCK+1
000004      REWIND LDATA
000006      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NTIME,1,5)
000024      DO 400 I=1,NLOAD
000026      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,I),F(1,I),B(1,I),XNE(I),XNM(I),
     1 NLOAD,NTIME,KDIMM1,6)
000053      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1)
     1 ,XNE(KDIMP1),XNM(KDIMP1),NLOAD,NTIME,1,6)
000104  400 CONTINUE
000107      END FILE LDATA
000111      REWIND ISTORE
000113      PRINT 1, NBLOCK
000121      RETURN
000122      END
```

```
      SUBROUTINE SEGMNT(ISECT)
000003  COMMON  E(17,201),     F(18,201),     B(18,201),     XNE(201),
     1        XNM(201),     KOUT(200),  KPRINT(200),ICOUNT(200,8),
     2      NBEGIN(200),      DRTH(16),   EFACTA(16),   EFACTB(16),
     3         MTH(17),      LCON(17),       EK(17),       SIG(17),
     4       FFACTA(17),    FFACTB(17),       TH(18),       STH(18),
     5          FK(18),     HEADER(8),     HEAD(8), OUTARR(21,4),
     6       FTEMP(18),    FTEMP2(18),   FXTRAP(18),     THETA(18),
     7       HTHETA(17)
C
000003  COMMON /INTEGER/     KUATA,     ISTORE,      NTIME,       LOAD,
     1          NLOAD,       LMAX,     LMAXP1,       KDIM,     KDIMP1,
     2          IRENUM,     NKOUT,      LDATA,     NBLOCK,      JDATA,
     3          JSTORE,    NSHIFT,     KDIMM1,     LMAXP2,      NPACK,
     4          IDATA
C
000003  COMMON /SAVE/     ES(17,20), FS(18,20), BS(18,20),   XNES(20),
     1        XNMS(20)
C
000003  GO TO (100,200,300,400), ISECT
C
C
C         READ IN FROM DISK LEFT MOST SEG TO START NEW LEFT MOST SEG AT
C         NEW DELTA T
C
C
000013  100 CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NTIME,1,7)
000031      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(2),XNM(2),
     1  NLOAD,NTIME,KDIMM1,7)
000060      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1)
     1  ,XNE(KDIMP1),XNM(KDIMP1),NLOAD,NTIME,1,7)
000113      RETURN
C
000114  200 IF (LOAD.EQ.0) LOAD=1
C
C         SAVE SEGMENT JUST COMPUTED ON DISK TO FREE MEMORY
C
C
000117      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NTIME,1,8)
000135      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(2),XNM(2),
     1  NLOAD,NTIME,KDIMM1,8)
C
000164      IF(LOAD.EQ.1) GO TO 220
000167      LOADM1=LOAD-1
000170      DO 210 L=1,LMAX
000172      ES(L,LOADM1)=E(L,2)
000175      FS(L,LOADM1)=F(L,2)
000201      BS(L,LOADM1)=B(L,2)
000204  210 CONTINUE
000206      FS(LMAXP1,LOADM1)=F(LMAAP1,2)
000212      BS(LMAXP1,LOADM1)=B(LMAAP1,2)
000216      XNES(LOADM1)=XNE(2)
000220      XNMS(LOADM1)=XNM(2)
C
C         LOAD IS INCREMENTED WHEN THE RANGE LOOP TERMINATES
C         NORMALLY. LOAD = THE NUMBER OF SEGMENTS ON DISK.
C
C
000221  220 LOAD=LOAD+1
000223      DO 230 L=1,LMAX
000224      E(L,1)=E(L,KDIMP1)
```

```
000227          F(L,1)=F(L,KDIMP1)
000232          B(L,1)=B(L,KDIMP1)
000235      230 CONTINUE
000237          F(LMAXP1,J)=F(LMAXP1,KDIMP1)
000243          B(LMAXP1,1)=B(LMAXP1,KDIMP1)
000246          XNE(1) = XNE(KDIMP1)
000250          XNM(1) = XNM(KDIMP1)
000251          IF(LOAD.LE.NLOAD) GO TO 250
                C
                C       NLOAD IS ADVANCED WHEN A NEW COMPLETE SEGMENT W/ KDIM MEMBERS
                C       IS ADDED TO THE SEGMENTS BEING HELD ON DISK. NLOAD IS = TO
                C       NUMBER OF SEGMENTS ON DISK + 1
                C
000254          NLOAD=NLOAD+1
000255          DO 240 K=2,KDIMP1
000257          XNE(K)=0.
000260          XNM(K)=0.
000261          F(LMAXP1,K)=0.
000264          B(LMAXP1,K)=0.
000267          DO 240 L=1,LMAX
000270          E(L,K)=0.
000273          F(L,K)=0.
000275      240 B(L,K)=0.
000304          RETURN
                C
000305      250 IF(NSHIFT.NE.2) GO TO 260
                C
                C       READ IN FROM DISK NEXT SEG TO COMP SAME SLG AT NEW DELTA T
                C
000306          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(2),XNM(2),
                1 NLOAD,NTIME,KDIMM1,7)
000335          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1)
                1 ,XNE(KDIMP1),XNM(KDIMP1),NLOAD,NTIME,1,7)
000370          RETURN
                C
000371      260 KDIMM2=KDIM-2
000373          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(2),XNM(2),
                1 NLOAD,NTIME,KDIMM2,7)
000422          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIM),F(1,KDIM),B(1,KDIM),
                1 XNE(KDIM),XNM(KDIM),NLOAD,NTIME,1,7)
000465          IF(LOAD.EQ.NLOAD) GO TO 275
000460          DO 270 L=1,LMAX
000462          E(L,KDIMP1)=ES(L,LOAD)
000467          F(L,KDIMP1)=FS(L,LOAD)
000474          B(L,KDIMP1)=BS(L,LOAD)
000502      270 CONTINUE
000504          F(LMAXP1,KDIMP1)=FS(LMAXP1,LOAD)
000512          B(LMAXP1,KDIMP1)=BS(LMAXP1,LOAD)
000517          XNE(KDIMP1)=XNES(LOAD)
000521          XNM(KDIMP1)=XNMS(LOAD)
000523          RETURN
000523      275 DO 280 L=1,LMAX
000525          E(L,KDIMP1)=0.0
000530          F(L,KDIMP1)=0.0
000532          B(L,KDIMP1)=0.0
000535      280 CONTINUE
000537          F(LMAXP1,KDIMP1)=0.0
000542          B(LMAXP1,KDIMP1)=0.0
```

```
000545          XNE(KDIMP1)=0.0
000546          XNM(KDIMP1)=0.0
000547          RETURN
                C
                C       SAVE RIGHT MOST SEG ON DISK, PREPARE TO STEP TIME AND READ IN
                C       EARLIER COMPUTED SEG AT START OF TIME LOOP
                C
000547        300 CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NTIME,1,8)
000565          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(2),XNM(2),
                1 NLOAD,NTIME,KDIMM1,8)
000614          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1)
                1 ,XNE(KDIMP1),XNM(KDIMP1),NLOAD,NTIME,1,8)
000647          REWIND ISTORE
000651          REWIND JSTORE
000653          I=ISTORE
000655          ISTORE=JSTORE
000656          JSTORE=I
000657          IF(LOAD.EQ.1) RETURN
000662          LOADM1=LOAD-1
000664          DO 35. L=1,LMAX
000666          ES(L,LOADM1)=E(L,2)
000671          FS(L,LOADM1)=F(L,2)
000675          BS(L,LOADM1)=B(L,2)
000700        350 CONTINUE
000702          FS(LMAXP1,LOADM1)=F(LMAXP1,2)
000706          BS(LMAXP1,LOADM1)=B(LMAXP1,2)
000712          XNES(LOADM1)=XNE(2)
000714          XNMS(LOADM1)=XNM(2)
000715          LOAD=1
000716          RETURN
000717        400 CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NTIME,1,8)
000735          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(2),XNM(2),
                1 NLOAD,NTIME,KDIMM1,8)
000764          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1)
                1 ,XNE(KDIMP1),XNM(KDIMP1),NLOAD,NTIME,1,8)
001017          END FILE JSTORE
001021          REWIND JSTORE
001023          RETURN
001024          END
```

```
      SUBROUTINE PRINTER
000002      COMMON  E(17,201),     F(18,201),     B(18,201),     XNE(201),
     1         XNM(201),     KOUT(200),  KPRINT(200),ICOUNT(200,8),
     2       NBEGIN(200),     DMTM(16),   EFACTA(16),   EFACTB(16),
     3          HTH(17),     ECON(17),       EK(17),       SIG(17),
     4       FFACTA(17),   FFACTB(17),       TH(18),       STH(18),
     5          FK(18),   HEADER(8),     HEAD(8), OUTARR(21,4),
     6       FTEMP(18),   FTEMP2(18),   FXTRAP(18),     THETA(18),
     7       HTHETA(17)
      C
      C       **NOTE** WHEN DIMENSIONS OF OUTARR ARE CHANGED,IROW CHECK
      C       SHOULD BE CHANGED
      C
000002      COMMON /INTEGER/      KDATA,     ISTORE,      NTIME,       LOAD,
     1           NLOAD,       LMAX,     LMAXP1,       KDIM,    KDIMP1,
     2          IRENUM,     NOUT,       LDATA,     NBLOCK,      JDATA,
     3          JSTORE,     NSHIFT,     KDIMM1,     LMAXP4,      NPACK,
     4           IDATA
      C
000002      COMMON /RANDT/           R,        ROC,       RPDR,     RPHDR,
     1          RPHDRC,     RSTART,        DR,        HDR,       LGM1,
     2            LGND,     LGNDP1,         K,        KP1,       KNUM,
     3          NFILES,     NSHIFT,      QCON,     EMOCON,    EMOBCN,
     4           CMUDR,     DTEPS,     XTRAP,          T,       TPDT,
     5           TPHDT,     THDUT,        DT,       AGND,       XGND,
     6        ORNG(200)
      C
000002       1 FORMAT(///3I10)
000002       9 FORMAT(1H1/////1X,8A1,,18X,*RANGE NO. *,I2,11X,*PAGE *,I3)
000002      10 FORMAT(//11X,*TIME NION=NSEC*,6X,*TIME TOT SIGMA*,6X,*TIME B PHI*,
     1  10X,*TIME E THETA*,17X,*IRENUM = *,I3,5X,*TIME E RADIAL*)
000002      11 FORMAT(/5X,4(5X,E15.8),,1X,E15.8)
000002      12 FORMAT(/3X,*NION = *,E15.8,15X,*CURRENT = *,E15.8)
000002      13 FORMAT(/2X,*THETA*,9X,*NSEC*,13X,*SIGMA TOTAL*,11X,*B PHI*,14X,
     1  *E THETA*,23X,*THETA/2*,9X,*E RADIAL*/)
000002      14 FORMAT(F4.3,2X,E15.8,3(5X,E15.8))
000002      15 FORMAT(1X,F8.3,7X,E15.8)
000002      16 FORMAT(F8.3,17X,3(5X,E15.8))
000002      22 FORMAT(2X,*GROUND*,I2,(1H-),*GROUND *,E15.8,5X,E15.8,1X,*GROUND*,
     1  I2,(1H-),F8.3,7X,E15.8)
000002      23 FORMAT(F8.3,37X,2(5X,E15.8))
      C
000002          OUTARR(1,1)=T
000004          OUTARR(1,2)=TPHDT-ROC
000006          OUTARR(1,3)=T-RPHDRC
000010          OUTARR(1,4)=TPDT-ROC
000011          OUTARR(2,1)=ORNG(KNUM)
000013          OUTARR(2,2)=F(LGM1,K)+XTRAP*(F(LGND,K)-F(LGM1,K))
000024          OUTARR(2,4)=B(LGM1,K)+XTRAP*(B(LGND,K)-B(LGM1,K))
000032          OUTARR(3,1)=TPHDT-RPHDRC
000034          OUTARR(3,2) = XNM(KP1)
000036          OUTARR(3,3)=XNE(KP1)
000040          OUTARR(3,4)=XJAY(R,TPDT)
000043          DO 31 L=1,LMAXP1
000044          LP3=L+3
000046          IF (L.GT.LGND) GO TO 300
000051          OUTARR(LP3,3)=SIG(L)
```

```
000053      300 OUTARR(LP3,4)=B(L,K)
000060          OUTARR(LP3,1)=F(L,K)
000063          IF (L.EQ.LMAXP1) GO TO 310
000064          OUTARR(LP3,2)=E(L,K)
000070      310 CONTINUE
000073          ICOUNT(KNUM,NFILES)=ICOUNT(KNUM,NFILES)+1
            C
000077          IF (KPRINT(KNUM).EQ.0) GO TO 370
000100          KPRINT(KNUM)=KPRINT(KNUM)-1
000102          IPAGE = 0
000102          DO 314 J=1,NFILES
000104          IPAGE = IPAGE + ICOUNT(KNUM,J)
000111      314 CONTINUE
000113          PRINT  9,HEADER,KNUM,IPAGE
000125          PRINT 10,IRENUM
000133          PRINT 11, OUTARR(1,2),OUTARR(1,3),OUTARR(1,4),OUTARR(3,1),OUTARR(1
                1 ,2)
000151          PRINT 12,XNM(KP1),OUTARR(3,4)
000161          PRINT 13
000165          PRINT 14, THETA(1),OUTARR(3,3),OUTARR(4,3),OUTARR(4,4),OUTARR(4,1)
000203          DO 320 M=1,LGM1
000205          L=M+4
000207          LMO=M+3
000211          PRINT 15, HTHETA(M),OUTARR(LMO,2)
000220          PRINT 19, THETA(M+1),OUTARR(L,3),OUTARR(L,4),OUTARR(L,1)
000234      320 CONTINUE
000237          PRINT 22, OUTARR(2,3),OUTARR(2,2),HTHETA(LGND),OUTARR(L,2)
000252          LMMO=LMAX-1
000254          DO 330 M=LGND,LMMO
000256          LPJ=M+4
000260          MP1=M+1
000262          PRINT 23, THETA(MP1),OUTARR(LP3,4),OUTARR(LP3,1)
000273          PRINT 15, HTHETA(MP1),OUTARR(LPJ,2)
000303      330 CONTINUE
000306          LST=LMAX+4
000310          PRINT 23, THETA(LMAX+1),OUTARR(LST,4),OUTARR(LST,1)
000321          PRINT 1, NSHIFT,NSHIFT,NTIME
000333      370 IF(IRENUM.LT.1) RETURN
            C
            C       WRITE OUTPUT DATA ON TAPE JDATA
000335          IROW=3
000336          ICOL=1
000337          I=3
000340          II=4
000341          J=1
000342          JJ=1
000343          LMAXP3=LMAX+3
000345          DO 400 IPACK=1,NPACK
000347          OUTARR(IROW,ICOL)=PACK(OUTARR(I,J),OUTARR(II,JJ))
000364          IROW=IROW+1
000365          IF(IROW.LE.21) GO TO 385
000367          IROW=1
000370          ICOL=ICOL+1
000371      385 I=I+2
000373          II=II+2
000374          IF(I.LT.LMAXP4) GO TO 400
000376          ITEST=I-LMAXP3
000400          GO TO (386,390,392), ITEST
```

```
000406      386 II=1
000407          JJ=JJ+1
000411          GO TO 400
000411      390 I=1
000412          J=J+1
000414          II=I+1
000415          JJ=JJ+1
000416          GO TO 400
000417      392 I=2
000420          J=J+1
000422      400 CONTINUE
000425          NPACKS=NPACK+2
000427          CALL DATAPE(OUTARR,NPACKS,2)
000431          RETURN
000432          END
```

```
      SUBROUTINE SHIFTY
      COMMON  E(17,201),     F(18,201),     B(18,201),     XNE(201),
     1        ...M(201),     KOUT(200),  KPRINT(200),ICOUNT(200,8),
     2     NBEGIN(200),      HTH(16),    EFACTA(16),    EFACTB(16),
     3         HTH(17),      ECON(17),       EK(17),       SIG(17),
     4      FFACTA(17),    FFACTB(17),       TH(18),       STH(18),
     5          FX(18),    HEADER(8),     HEAD(8), OUTARR(21,4),
     6       FTEMP(18),   FTEMP2(18),  FXTRAP(16),     THETA(18),
     7       HTHETA(17)
C
      COMMON /PARAMS/          C,     COULOM,       CMU,       EPS,
     1        EPSGND,       SOND,        RMO,    ATTACH,      XELEN,
     2        XIOMOB,     XELMGE,     XIOCOM,       ALF
C
      COMMON /INTEGER/      KDATA,     ISTORE,     NTIME,      LOAD,
     1         NLOAD,       LMAX,     LMAXP1,      KDIM,    KDIMP1,
     2        IRENUM,      NKOUT,      LDATA,    NBLOCK,     JDATA,
     3        JSTORE,     NSHIFT,     KDIMM1,    LMAXP4,     NPACK,
     4         IDATA
C
      COMMON /RANDT/           R,        RUC,       RPOR,     RPHOR,
     1        RPHORC,     RSTART,         UR,        HOR,      LGM1,
     2          LGND,     LGNDP1,          K,        KP1,      KNUM,
     3        NFILES,     NSHIFT,       JCON,    EMOCON,    EMOBCN,
     4         CMUOR,     DTEPS2,      XTRAP,          T,      TPDT,
     5          TPHDT,     TMOUT,         DT,       AGND,      XGND,
     6       ORNG(200)
C
      COMMON /SAVE/     ES(17,20),  FS(18,20),  BS(18,20),  XNES(20),
     1        XNMS(20)
C
      R=RSTART+HOR
      RPHOR=R+HOR
      RPOR=R+OR
      G=CUM*XJAY(RPHOR,TMHOT)
      XNEAT=XNE(2)
      XNMAT=XNM(2)
      TEMP=ATTACH+ALF*(XNEAT+XNMAT)
      ARG=DT*TEMP
      EXPON=EXP(-ARG)
      IF(ARG.GE.1.E-7) GO TO 100
      FACTOR=ARG
      GO TO 150
  100 FACTOR=1.-EXPON
  150 XNE(1)=EXPON*XNEAT+FACTOR*G/TEMP
      TEMP=XIOCOM*(XNEAT+XNMAT)
      ARG=DT*TEMP
      EXPON=EXP(-ARG)
      IF(ARG.GE.1.E-7) GO TO 200
      FACTOR=ARG
      GO TO 250
  200 FACTOR=1.-EXPON
  250 XNM(1)=EXPON*XNMAT
      IF(TEMP.LE.0.) GO TO 300
      XNM(1)=XNM(1)+FACTOR*ATTACH*XNEAT/TEMP
  300 CONTINUE
      CXAXPX=COULOM*XIOMOB*(2.*XNM(1)+XNE(1))
```

```
000075          CEAN=COULOM*EMOCON*ANE(1)
000077          DO 6 L=1,LMAXP1
000101          FK(L)=F(L,2)
000103          IF(L.GE.LGNDP1) GO TO 560
000105          IF(L.GT.1) GO TO 350
000110          EVAL=.5*(E(1,1)+E(1,2))
000113          GO TO 400
000113      350 LM1=L-1
000115          EVAL=.5*(EFACTA(LM1)*(E(L,1)+E(L,2))+EFACTB(LM1)*(E(LM1,1)+E(LM1,2
                1 )))
000125      400 SIG(L)=CEAN*(EMOBFN(SQRT(EVAL**2+FK(L)**2)*EMOBCN))+CXXXPX
000141          TEMP=SIG(L)*DTEPS
000143          IF(TEMP.LE.15.) GO TO 450
000145          X=0.0
000146          A=1.0/SIG(L)
000150          GO TO 550
000151      450 X=EXP(-TEMP)
000155          A=(1.0-X)/SIG(L)
000160          GO TO 550
000161      500 IF(L.GT.LGNDP1) GO TO 550
000165          X=AGND
000166          A=AGND
000167      550 F(L,1)=X*FK(L)-A*(RPMDR*B(L,2)-R*B(L,1))/RPMDR/CMUDR
            C     IF(ABS(F(L,1)).LT.1.E-63) F(L,1)=0.
000200      600 CONTINUE
000203          B(LMAXP1,1)=B(LMAXP1,2)
000204          DO 650 L=1,LMAX
000206          E(L,1)=E(L,2)
000210          B(L,1)=B(L,2)
000211      650 CONTINUE
000213          DO 700 K=2,KDIM
000215          KP1=K+1
000217          XNE(K)=XNE(KP1)
000221          XNM(K)=XNM(KP1)
000222          F(LMAXP1,K)=F(LMAXP1,KP1)
000227          B(LMAXP1,K)=B(LMAXP1,KP1)
000234          DO 700 L=1,LMAX
000236          E(L,K)=E(L,KP1)
000243          F(L,K)=F(L,KP1)
000250          B(L,K)=B(L,KP1)
000256      700 CONTINUE
000262          IF(LOAD.GT.0) GO TO 800
000265          XNE(KDIMP1)=0.
000266          XNM(KDIMP1)=0.
000267          F(LMAXP1,KDIMP1)=0.
000272          B(LMAXP1,KDIMP1)=0.
000275          DO 750 L=1,LMAX
000276          E(L,KDIMP1)=0.
000301          F(L,KDIMP1)=0.
000303          B(L,KDIMP1)=0.
000306      750 CONTINUE
000310          RETURN
000310      800 XNE(KDIMP1)=XNES(LOAD)
000313          XNM(KDIMP1)=XNMS(LOAD)
000314          F(LMAXP1,KDIMP1)=FS(LMAAP1,LOAD)
000321          B(LMAXP1,KDIMP1)=BS(LMAAP1,LOAD)
000326          DO 850 L=1,LMAX
000330          E(L,KDIMP1)=ES(L,LOAD)
```

```
000335          F(L,KDIMP1)=FS(L,LUAU)
000342          B(L,KDIMP1)=BS(L,LUAU)
000350      850 CONTINUE
000352          RETURN
000352          END
```

```
                      SUBROUTINE IOSAVE(ANES,ANMS,ES,FS,BS,E,F,B,XNE,XNM,JSET,NMIN,KSET,
                     1 IUSE)
000021                COMMON EE(17,201),    FF(18,201),    BB(18,201),    XXNE(201),
                     1       XXNM(201),    KOUT(200),  KPRINT(200),ICOUNT(200,8),
                     2     NBEGIN(200),    DMIN(16),   EFACTA(16),   EFACTB(16),
                     3         MIN(17),    ECON(17),       E<(17),       SIG(17),
                     4      FFACTA(17),  FFACTB(17),       TH(18),       STH(18),
                     5          FK(18),   HEADER(8),     HEAD(8), OUTARR(21,4),
                     6       FTEMP(18),  FTEMP2(18),  FXTRAP(18),    THETA(18),
                     7      HTHETA(17)
                      C
000021                COMMON /INTEGER/      KDATA,     ISTORE,      NTIME,       LOAD,
                     1          NLOAD,      LMAX,     LMAAP1,       KDIM,     KDIMP1,
                     2          IRENUM,     NKOUT,      LDATA,     NBLOCK,      JDATA,
                     3          JSTORE,    NSHIFT,     KDIMM1,     LMAXP4,      NPACK,
                     4           IDATA
                      C
000021                COMMON /RANDT/            K,        KOC,       RPDR,      RPHDR,
                     1        RPHDRC,     KSTART,         DK,        HDR,       LGM1,
                     2          LGND,     LGNDP1,          K,        KP1,       KNUM,
                     3         NFILES,    NSHFTT,       QCON,     EMOCON,    EMOBCN,
                     4          CMUDR,      DTEPS,     XTRAP,          T,       TPDT,
                     5          TPHDT,      THDT,         DT,       AGND,       XGND,
                     6       ORNG(200)
                      C
000021                DIMENSION XNES(JSET),XNMS(JSET),ES(17,JSET),FS(18,JSET),BS(18,JSET
                     1 ),E(17,KSET),F(18,KSET),B(18,KSET),XNE(KSET),XNM(KSET),DUM1(17),
                     1 DUM2(18)
000021                GO TO (1,0,200,300,400,500,600,700,800,900), IUSE
000036              100 READ(KDATA) ICOUNT,KPRINT,NMIN,LOAD,NLOAD,NBLOCK,NSHFTT,RSTART,
                     1 FTEMP,FXTRAP,XNES,XNMS,ES,FS,BS
000130              200 READ (KDATA ) E,F,B,XNE,XNM
000200                  WRITE(ISTORE) E,F,B,XNE,XNM
000260                  RETURN
000251              300 WRITE(LDATA)ICOUNT,KPRINT,NMIN,LOAD,NLOAD,NBLOCK,NSHFTT,RSTART,
                     1 FTEMP,FXTRAP,XNES,XNMS,ES,FS,BS
000347              400 READ (JSTORE) E,F,B,XNE,XNM
000417                  WRITE(LDATA ) E,F,B,XNE,XNM
000467                  RETURN
000470              500 WRITE(LDATA)ICOUNT,KPRINT,NMIN,LOAD,NLOAD,NBLOCK,NSHFTT,RSTART,
                     1 FTEMP,FXTRAP,XNES,XNMS,ES,FS,BS
000566              600 IF(JSET.EQ.1) GO TO 650
000574                  READ (ISTORE) E,F,B,XNE,XNM
000641              650 WRITE(LDATA ) E,F,B,XNE,XNM
000711                  RETURN
000712              700 READ(ISTORE) E,F,B,XNE,XNM
000762                  RETURN
000763              800 WRITE(JSTORE) E,F,B,XNE,XNM
001033                  RETURN
001034              900 READ(ISTORE) DUM1,E,DUM2,F,DUM2,B,DUM3,XNE,DUM3,XNM
001117                  RETURN
001120                  END
```

```
                      SUBROUTINE DATAPE(OUTARR,NPACKS,IOPT)
                      COMMON /INTEGER/      KDATA,    ISTORE,     NTIME,      LOAD,
000006               1          NLOAD,       LMAX,     LMAXP1,      KDIM,    KDIMP1,
                     2          IRENUM,     NKOUT,      LDATA,    NBLOCK,     JDATA,
                     3          JSTORE,    NSHIFT,    KDIMM1,    LMAXP2,     NPACK,
                     4            IDATA
000006                DIMENSION OUTARR(NPACKS)
                     C
000006                GO TO (100,200), IOPT
000013           100  READ(IDATA) OUTARR
000022                IF(EOF,IDATA) 250,200
000027           200  WRITE(JDATA) OUTARR
000036                RETURN
000037           250  IOPT=0
000042                RETURN
000042                END
```

```
      BLOCK DATA
C
C *** TEST PROBLEM                    ***
C
000002      COMMON /PARAMS/          C,      COULOM,       CMU,      EPS,
     1           EPSGND,       SOND,        RMU,      ATTACH,     XELEN,
     2           X10MOB,     XELMGE,     X10COM,        ALF
C
000002      DATA                     C,      COULOM,       CMU,      EPS,
     1           EPSGND,       SOND,        RMU,      ATTACH,     XELEN,
     2           X10MOB,     XELMGE,     X10COM,        ALF/
     3                         3.E8,     1.6E-19, 12.566E-7,.88416E-11,
     4   14.14556E-11,         5.E-3,       1.293,       1.E8,      -1.4,
     5          2.7E-4,         4.51,     2.3E-12,     4.5E-13/
000002      END
```

```
                                                                  COMPASS - VER 2.     06/28/71  11.30.39.     PAGE    2
PACK
                                                                                                        00000000
                                           IDENT PACK                                                   00000010
                                           ENTRY PACK                                                   00000020
                                           ENTRY UNPACK                                                 00000030
                                           VFD  42/0LPACK+18/2                                          00000040
     0  20015313000000000004     PACK      DATA 0                                                       00000050
     1  00000000000000000002               SA1  B1                                                      00000060
     2  56110                              SA4  ZLOHF                                                   00000070
            5140000012 +                   BX1  X1*X4                                                   00000080
                      11114                SA2  B2                                                      00000090
     3  56620                              LX2  36B                                                     00000100
            20236                          SA3  ZUPHF                                                   00000110
                 5130000011 +              BX2  X2*X3                                                   00000120
     4  11223                              BX6  X1+X2                                                   00000130
            12612                          EQ   PACK                                                    00000140
                 0400000001 +  UNPACK      DATA 0                                                       00000150
     5  00000000000000000002               SA1  B1                                                      00000160
     6  56110                              BX6  X1                                                      00000170
            10611                          BX7  X1                                                      00000180
                 10711                     AX6  36B                                                     00000190
                      21630                LX6  36B                                                     00000200
     7  20636                              SA6  B2                                                      00000210
            56620                          LX7  36B                                                     00000220
                 20736                     AX7  36B                                                     00000230
                      21730                LX7  36B                                                     00000235
    10  20736                              SA7  B3                                                      00000240
            56730                          EQ   UNPACK                                                  00000250
                 0400000005 +                                                                           
                         11 +  ZUPHF       LIT  0000000000777777777B                                    00000260
                         12 +  ZLOHF       LIT  7777777777000000000B                                    00000280
                                           END
    13                45003    STORAGE USED               29 STATEMENTS         4 SYMBOLS
                               6400 ASSEMBLY              0.238 SECONDS        10 REFERENCES
```

B. ORESTES

1. Description of Modifications

Program ORESTES[1] received two minor modifications since it was last documented. All input/output (I/O) operations using tape or drum were revised in order to cut running time and to make more efficient use of data tapes.

The running time was cut by replacing all lengthy I/O lists, which used implied DO loops, with a variably dimensioned subroutine (IOSAVE) which can write or read any length array as a single operation. This modification should produce up to 90% reduction in running time for the operations which it affects.

More efficient use of data tapes was accomplished by revising the contents of the data output array (OUTARR). By removing some unnecessary data items from the array and repositioning other items, the dimensions of the array were reduced from 27x11 to 25x9 resulting in a decrease of 72 words. The words in the reduced array were then packed two into one by an assembly language packing routine (PACK). The net result was to reduce the number of words written to tape, at a given time step and range step, from 297 to 114. If the number of angles used for a given run is less than 24, the number of words written to tape will be reduced even more.

---

1. Jones, D. L. and D. H. Stump, ELECTRA, ORESTES, and Supporting Graphic Display Codes, U.S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Virginia, February 1971, pp. 37-69.

2. Program Listing

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

              PROGRAM ORESTES(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE7,TAPE8,
             1 TAPE5=INPUT,TAPE6=OUTPUT)
          C
000003         COMMON         E(23, 251),    F(24, 251),     H(24, 251),   XNE(24,251),
             1              XNM(24,251),      CEX(24),     CA2XPX(24),    CXXXPX(24),
             2                CEXN(24),      KOUT(16),     KPRINT(16),ICOUNT(16,10),
             3             NBEGIN(16),       DMTH(23),     EFACTA(23),    EFACTB(23),
             4                HTH(23),       ECON(23),         EK(23),       SIG(24),
             5              FFACTA(23),     FFACTB(23),        TH(24),       STH(24),
             6                 FK(24),      HEADER(8),       HEAD(8),  OUTARR(25,9),
             7               FTEMP(24),     FTEMP2(24),     FXTRAP(24),     THETA(24),
             8              HTHETA(23)
          C
000003         COMMON/SOURCE/          XJ1,          XJ2,            XJ3,          XJ4,
             1                          T0,           T1,             T2,           T3,
             2                       XJRUN,       CLAMDA
          C
000003         COMMON /PARAMS/            C,       COULOM,           CMU,          EPS,
             1              EPSGND,       SGND,         RHO,       ATTACH,        XELEN,
             2              XIOMOB,     XELRGE,      XIOCON,          ALF,         QCON,
             3              ITHETA,     IHTHET,       SCALE
          C
000003         COMMON /INTEGER/       KDATA,       ISTORE,          NTIME,         LOAD,
             1               NLOAD,       LMAX,       LMAXP1,          KDIM,       KDIMP1,
             2              IRENUM,       NKOUT,       LDATA,        NBLOCK,        JDATA,
             3              JSTORE,      NSHIFT,      KDIMM1,        LMAXP2,        NPACK,
             4                IREM,       IDATA
          C
000003         COMMON /RANDT/              R,          RUC,          RPDR,        RPHDR,
             1              RPHDRC,      KSTART,          DR,           HDR,         LGM1,
             2                LGND,      LGNDP1,           R,           KP1,         KNUM,
             3              NFILES,      NSHFTT,                      EMUCON,       EMUBCN,
             4               CMUDR,       DTEPS,       XTRAP,              T,         TPDT,
             5               TPMDT,       TMMDT,          DT,          AGND,         XGND,
             6           ORNG(16),   XJRR(24),      XJT(24),         Q(24)
          C
000003         COMMON /SAVE/      ES(23,20),  FS(24,20),  BS(24,20),XNES(24,20),
             1       AMPS(24,20)
          C
000003         DIMENSION      NSTOP(16),     NSTEP(16),       LAB(8)
          C
000003         DATA IDATA,JDATA,KDATA,LDATA,ISTORE,JSTORE,NSTEP,NSTOP
             1      /      1,       2,       3,      4,        7,       8,                    NSTEP
             2             2,     20,     200,   2000,    12*0,                              /NSTOP
             3           200,   1000,   1000,100000,    12*0
000003         DATA LAB/8HNKOUT   =,8HKDIM    =,8HNMIN    =,8HNMAX     =,8HNSTEPS =,
             1        8HNSTART =,8HLGND    =,8HLMAX    =/
          C
000003        1 FORMAT(I10,7F10.0/(8F10.0))
000003        2 FORMAT(8I10)
000003        3 FORMAT(16I5)
000003        5 FORMAT(1H1///2X,6HRMIN =,F8.3,10X,4HDR =,F7.4,10X,4HDT =,E10.3,
             1   7X,6HRCAL =,F8.3,7X,7HSCALE =,F10.2,9X,5HRHO =,F7.4)
000003        6 FORMAT(///19X,4(A8,I6,10X))
000003        7 FORMAT(///(19X,8HKOUT    =,I6,18X,I5,18X,I6,18X,I6))
000003        8 FORMAT(*0IRENUM ON TAPE =*,I2,*, IRENUM ON INPUT CARD =*,I2)
000003       24 FORMAT(/10H NBLOCK =  ,I5)
```

```
000003      25  FORMAT (*,ERROR...NUMBER OF OUTPUT RANGES = *,I5)
000003      26 FORMAT(24,*RMIN READ IN IS TOO SMALL RCAL = *,F8.3)
000003     610 FORMAT(1H1,8A10)
000003     611 FORMAT(*0*... OUTPUT DATA FILES ...*)
000003     612 FORMAT(*0FILE*,I3,6A,*RANGE NO*,5X,*RANGE*,5X,*NO OF RECS*)
000003     613 FORMAT(1H ,I21,F10.6,I15)
000003     614 FORMAT(*0NEXT VALUE OF I IS*,E16.8)
000003     615 FORMAT(*0ELECTRA...NORMAL TERMINATION*)
000003     999 FORMAT(8F10.0)
000003    1000 FORMAT(7A10,A8,I2)
000003    1013 FORMAT(1H1,8A10)
000003    1017 FORMAT(*0ELECTRA...ERROR TERMINATION*)
000003    1027 FORMAT(*0HEADER FOR FILE *,I2,* ON TAPE DOES NOT CHECK*)
000003    1028 FORMAT(*0HEADER ON TAPE= *,8A10)
          C
000003         IERR=0
000004         DO 1 0 I=1,10
000006         DO 100 J=1,16
000007         COUNT(J,I)=0
000012     100 CONTINUE
000016         DO 101 I=1,25
000017         DO 101 J=1,9
000020         OUTARR(I,J)=0.
000024     101 CONTINUE
000030         DO 105 I=1,16
000031         KPRINT(I)=0
000032     105 CONTINUE
000034         LOAD=0
000035         NLOAD=1
000036         NTIME=0
000037         NBLOCK=0
000040         NFILES=1
000041         NSHIFT=0
000042         RESTRT=1.
000043         REWIND ISTORE
000045         REWIND JSTORE
000047         READ 999,XJ1,XJ2,XJ3,XJ4,T0,T1,T2,T3,XJRUN,CLAMDA
000077         READ 1000,HEADER,IHENJM
000107         READ(5,999) DR,DT,TS,SEC,RMIN,RMO,SCALE
000131         READ(5,1) ITHETA,(THETA(I),I=1,ITHETA)
000146         READ(5,1) IHTHET,(HTHETA(I),I=1,IHTHET)
000163         READ(5,2) NSEG,KDIM,NMAX,NSTEPS,NSTART,LGND,LMAX
000205         READ(5,3) NKPRNT,(KPRINT(I),I=1,NKPRNT)
000222         IF (SEC.LE.0.) SEC=1000.
000225         IF(NSTART.LE.0) NSTART=2
000230         IF(NSTEPS.LE.0) NSTEPS=9
000233         READ (5,2) NKOUT
000241         IF (NKOUT.LE.0) GO TO 121
000243         READ (5,999) (ORNG(I),I=1,NKOUT)
000255         GO TO 140
000256     121 M=1
000257         READ (5,1) NDROUT, ORNG(1)
000267         TEMPOR=ORNG(1)
          C
000271         DO 130 JJ=1,NDROUT
000272         READ (5,1) NROUT, DROUT
000301         DO 125 I=1,NROUT
000303         K=1+M
```

```
000305          TEMPOR=TEMPOR+DROUT
000307          ORNG(K)=TEMPOR
000311      125 CONTINUE
000313          M=M+NROUT
000314      130 CONTINUE
000316          IF (K.LE.16) GO TO 135
000321          PRINT 25,K
000326          STOP
000330      135 NKOUT=K
000332      140 LGNDP1=LGND+1
000334          LGM1=LGND-1
000335          LMAXP1=LMAX+1
000337          LMAXP2=LMAX+2
000341          NPACK=(LMAXP2*9)/2-1
000344          IREM=MOD(LMAXP2*9,2)
000350          KDIMP1=KDIM+1
000351          KDIMM1=KDIM-1
000353          DELMIN=3.14
000354          DO 1225 L=1,LMAXP1
000356          TH(L)=.01745329*THETA(L)
000360          STH(L)=SIN(TH(L))
000364          IF(L.EQ.1) GO TO 1224
000366          DELTHE=TH(L)-TH(L-1)
000370          IF(DELMIN.GT.DELTHE) DELMIN=DELTHE
000373          IF (L.EQ.LMAXP1) GO TO 1225
000375     1224 HTH(L)=.01745329*HTHETA(L)
000400     1225 CONTINUE
000403          IF(LGND.GT.LMAXP1) GO TO 122
000406          XTRAP=COS(TH(LGM1))/(COS(TH(LGM1))-COS(TH(LGND)))
000421      122 RCAL=C*DT/(DELMIN*SQRT(1.-(C*DT/OR)**2)) +1.E-5
000433          IF(RMIN.GE.RCAL) GO TO 1227
000436          IF(RMIN.LE.0.) GO TO 1226
000437          PRINT 26, RCAL
000445          STOP
000447     1226 RMIN=RCAL
000451     1227 NMIN=RMIN/(C*DT)+1.
000456          RSTART=RMIN
000457          IF (IRENUM.LE.0) GO TO 95
000460          REWIND IDATA
000462          REWIND JDATA
000464          REWIND KDATA
000466          REWIND LDATA
                C
                C       READ THE RESTART TAPE
                C
000470          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NMIN,1,1)
000506          DO 1036 J=1,NLOAD
000510          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(1,2),XNM(1
                1 ,2),NLOAD,NMIN,KDIMM1,2)
000535          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1)
                1 ,XNE(1,KDIMP1),XNM(1,KDIMP1),NLOAD,NMIN,1,2)
000567      1036 CONTINUE
                C
000572          REWIND ISTORE
                C
000574          READ (IDATA) NFILES,NKOUT,LMAX
000605          IF(NFILES.EQ.IRENUM) GO TO 1024
000607          WRITE(6,8) NFILES,IRENUM
```

```
000617          STOP
000621     1024 NFILES=IRENUM+1
000623          WRITE (JDATA) NFILES,NKOUT,LMAX
                C
                C        DATA TAPE SETUP LOOP
                C
000634          READ(IDATA) HEAD,IRNUMT
000643          DO 1025 J=1,4
000645          IF (HEAD(J).NE.HEADER(J)) GO TO 1026
000647     1025 CONTINUE
000651          GO TO 1029
                C
                C        THE HEADER READ IN DOES NOT MATCH HEAD ON DATA TAPE
                C        PRINT HEADER,HEAD EXIT
                C
000652     1026 PRINT 1013,HEADER
000660          PRINT 1027,I
000656          PRINT 1028,HEAD
000674          PRINT 1017
000700          STOP
000702     1029 WRITE(JDATA) HEAD,IRENUM
000711          READ(IDATA) THETA,MIMETA
000720          WRITE(JDATA) THETA,MIMETA
000727          NPACKS=NPACK+2
000731          IF(IREM.NE.0) NPACKS=NPACKS+1
000734          IOPT=1
000735     1031 CALL DATAPE(OUTARR,NPACKS,IOPT)
000740          IF(IOPT) 1031,66,1031
                C
000742        65 IF (IRENUM.LT.0) GO TO 66
000744          REWIND JDATA
000746          WRITE (JDATA) NFILES,NKOUT,LMAX
000757          WRITE(JDATA) HEADER,IRENUM
000766          WRITE(JDATA) THETA,MIMETA
000775        66 IF(IRENUM.GT.0) GO TO 70
                C
                C        INITIALIZE ARRAYS TO ZERO ON FIRST PASS
                C
001000          DO 68 K=1,KDIMP1
001001          XNE(LMAXP1,K)=0.
001004          XNM(LMAXP1,K)=0.
001007          F(LMAXP1,K)=0.
001012          B(LMAXP1,K)=0.
001014          DO 68 L=1,LMAX
001016          XNE(L,K)=0.
001021          XNM(L,K)=0.
001024          E(L,K)=0.
001027          F(L,K)=0.
001031          B(L,K)=0.
001034        68 CONTINUE
001040          DO 69 L=1,LMAXP1
001042          SIG(L)=0.0
001043          FTEMP(L)=0.0
001044        69 CONTINUE
001046        70 DO 1235 L=1,LMAX
001050          LP1=L+1
001052          ECON(L)=CMU*(TH(LP1)-TH(L))*SIN(HTH(L))
001060          FFACTA(L)=(HTH(L)-TH(L))/(TH(LP1)-TH(L))
```

```
001066          FFACTS(L)=1.-FFACTA(L)
001071          IF (L.EQ.LMAX) GO TO 1235
001072          DHTH(L)=HTH(LP1)-HTH(L)
001075          EFACTA(L)=(TH(LP1)-HTH(L))/DHTH(L)
001101          EFACTB(L)=1.-EFACTA(L)
001103     1235 CONTINUE
001106          IF(LGND.LE.LMAX) ECON(LGND)=.5*ECON(LGND)
001113          CDT=C*DT
001115          HDT=.5*DT
001117          HDR=.5*DR
001121          DO 1245 K=1,NKOUT
001122          KOUT(K)=(ORNG(K)+HDR-RMIN)/DR+.5
001130          R=RMIN-HDR+KOUT(K)*DR
001135          NBEGIN(K)=R/CDT
001140     1245 CONTINUE
001142          WRITE(6,5) RMIN,DR,DT,RCAL,SCALE,RMO
001161          WRITE(6,6) LAB(1),NKOUT,LAB(2),KDIM,LAB(3),NMIN,LAB(4),NMAX
001205          WRITE(6,6) LAB(5),NSTEPS,LAB(6),NSTART,LAB(7),LGND,LAB(8),LMAX
001231          WRITE(6,7) (KOUT(I),I=1,NKOUT)
001244          EMUCON=1.293/RMO
001246          EMUBCN=EMUCON/76000.
001250          TEMP=XELEN/34.E-6
001252          QCON=-TEMP/(COULOM*ALROE)
001255          DTEPS = DT/EPS
001257          DTEPSG = DT/EPSGND
001261          XGND = EXP(-SGND*DTEPSG)
001265          AGND = (1.0-XGND)/SGND
001270          CMUDR = CMU*DR
001272          DTDR = DT/DR
001273          C1=C*DT/DR
001275          C2=.5*C1*RMIN/DR+NSEG*KDIM-.900000
                C
                C       TIME LOOP
                C
001304          DO 410 N=NMIN,NMAX
001306          T=N*DT
001311          NTIME=N
001312          CALL SECOND(TIME)
001314          IF(TIME.GE.TS) GO TO 420
001317          IF (TIME.LT.RESTRT*SEC) GO TO 103
001322          CALL RESTART
001323          RESTRT=RESTRT+1.
001325      103 IF(LOAD.NE.0) CALL SEGMENT(1)
001330      102 TMMDT=T-HDT
001332          TPMDT=T+HDT
001334          TPDT=T+DT
001336          NSHIFT=C1*N-NSHFTT-62
001344          IF(NSHIFT.LT.0) NSHIFT=0
001346          NSHFTT=NSHFTT+NSHIFT
001350          DO 2005 L=1,LMAXP1
001351          FK(L)=F(L,1)
001353          FTEMP2(L)=FTEMP(L)
001354          FTEMP(L)=F(L,2)
001356     2005 CONTINUE
001360          IF(NSHFTT.EQ.0) GO TO 104
001361          IF(NSHIFT.EQ.0) GO TO 2010
001362          CALL SHIFT
001363          DO 2009 L=1,LMAXP1
```

```
001365          FXTRAP(L)=3.*F(L,1)-3.*FTEMP(L)+FTEMP2(L)
001375     2009 CONTINUE
001400          GO TO 120
001400     2010 DO 2011 L=1,LMAXP1
001402          F(L,1)=FXTRAP(L)
001405     2011 CONTINUE
001407      104 RSTART=RMIN+NSHFTT*DR
001413          R=RSTART-HDR
001415          CALL XJAY(TMHDT,RSTART,TMHDT,RSTART,XJRM,XJT,Q)
001423          DO 110 L=1,LMAXP1
001425          XNEX=XNE(L,1)
001427          TEMP=ATTACH+ALF*(XNEX+XNM(L,1))
001433          ARG=DT*TEMP
001434          EXPON=EXP(-ARG)
001440          IF(ARG .GE. 1.E-7) GO TO 16
001442          XNE(L,1)=(1.0-ARG)*XNEX+DT*Q(L)
001450          GO TO 17
001450       16 FACTOR = 1. - EXPON
001452          XNE(L,1)=EXPON*XNEX+FACTOR*Q(L)/TEMP
001457       17 TEMP=XIOCOM*(XNEX+XNM(L,1))
001463          ARG = DT*TEMP
001464          EXPON = EXP(-ARG)
001470          IF(ARG .GE. 1.E-7) GO TO 18
001472          XNM(L,1)=XNM(L,1)*(1.0-ARG)+DT*ATTACH*XNEX
001501          GO TO 21
001501       18 FACTOR = 1. - EXPON
001503       20 XNM(L,1)=EXPON*XNM(L,1)
001506          IF(TEMP .LE. 0.0) GO TO 21
001507          XNM(L,1)=XNM(L,1)+FACTOR*ATTACH*XNEX/TEMP
001514       21 CONTINUE
001514      110 CONTINUE
001517      120 LOADM1=LOAD-1
001521          IF(LOADM1.LT.0) LOADM1=0
                C
                C        RANGE LOOP
                C
001523          DO 370 K=1,KDIM
001525          R=K*DR
001527          RMHDR=R-HDR
001531          RPHDR=R+HDR
001532          RPDR=R+DR
001533          RPHDRC=RPHDR/C
001535          ROC=R/C
001536          DTOR=DT/R
001540          DTORDR=DTOR/DR
001542          IF (RPHDR.GE.C*TMHDT) GO TO 380
001545          KP1=K+1
001547          CALL XJAY(TMHDT,RPHDR,T,R,XJRM,XJT,Q)
001556          DO 1115 L=1,LMAXP1
001560          XNEX=XNE(L,KP1)
001564          TEMP=ATTACH + ALF*(XNEX + XNM(L,KP1))
001571          ARG = DT*TEMP
001573          EXPON = EXP(-ARG)
001576          IF(ARG .GE. 1.E-7) GO TO 1111
001600          XNE(L,KP1)=(1.0-ARG)*XNEX+DT*Q(L)
001611          GO TO 1112
001611     1111 FACTOR = 1. - EXPON
001613          XNE(L,KP1)=EXPON*XNEX + FACTOR*Q(L)/TEMP
```

```
001622    1112 TEMP=XIOCOM*(XNEX + XNM(L,KP1))
001630         ARG = DT*TEMP
001631         EXPON = EXP(-ARG)
001635         IF(ARG .GE. 1.E-7) GO TO 113
001637         XNM(L,KP1)=XNM(L,KP1)*(1.0-ARG)+DT*ATTACH*XNEX
001650         GO TO 1115
001651     113 FACTOR = 1. - EXPON
001653     114 XNM(L,KP1)=EXPON*XNM(L,KP1)
001660         XNM(L,KP1)=XNM(L,KP1) + FACTOR*ATTACH*XNEX/TEMP
001666    1115 CONTINUE
001671         DO 1120 L=1,LMAXP1
001672         CEXN(L)=COULOM*EMOCUN*XNE(L,KP1)
001677         CX2XPX(L)=COULOM*XIOMOB*(2.*XNM(L,KP1) + XNE(L,KP1))
001707         XSEC=0.5*(XNE(L,K)+XNE(L,KP1))
001715         XION=0.5*(XNM(L,K)+XNM(L,KP1))
001723         CEX(L)=COULOM*EMOCUN*XSEC
001726         CX2XPX(L)=COULOM*XIOMOB*(2.*XION + XSEC)
001733    1120 CONTINUE
              C
              C        THETA LOOP E
              C
001735         DO 185 L=1,LMAX
001737         EK(L)=E(L,K)
001743         LP1 = L+1
001745         IF (L.GE.LGND) GO TO 150
001747         FVAL=.5*(FFACTA(L)*(F(LP1,KP1)+F(LP1,K))+FFACTB(L)*(F(L,KP1)+F(L,K
              1))
001770         CEXMT=FFACTA(L)*CEX(LP1) + FFACTB(L)*CEX(L)
001775         CX2XMT=FFACTA(L)*CX2XPX(LP1) + FFACTB(L)*CX2XPX(L)
002001         SIGMA=CEXMT*(EMOBFN(SQRT(FVAL**2 + EK(L)**2)*EMOBCN)) + CX2XMT
002013         TEMP = SIGMA*DTEPS
002015         IF (TEMP.LE.15.) GO TO 142
002017         X = 0.0
002020         A = 1.0/SIGMA
002022         GO TO 180
002022     142 IF (TEMP.GE.1.E-5) GO TO 145
002025         X=1.-TEMP+.5*TEMP**2
002031         A=(1.-.5*TEMP)*DTEPS
002034         GO TO 180
002034     145 X=EXP(-TEMP)
002040         A = (1.0-X)/SIGMA
002043         GO TO 180
002043     150 IF (L.GT.LGND) GO TO 170
002047         SIGMA=CEX(L)*(EMOBFN(SQRT(.25*((EK(L-1) + EK(L))**2 + (FK(L) +
              1 F(L,KP1))**2))*EMOBCN)) + CX2XPX(L)
002067         TEMP = SIGMA*DTEPS
002071         IF (TEMP.LE.15.) GO TO 162
002073         XAIR = 0.0
002074         AAIR = 1.0/SIGMA
002076         GO TO 165
002076     162 IF (TEMP.GE.1.E-5) GO TO 163
002101         XAIR=1.-TEMP+.5*TEMP**2
002105         AAIR=(1.-.5*TEMP)*DTEPS
002110         GO TO 165
002110     163 XAIR=EXP(-TEMP)
002114         AAIR = (1.0-XAIR)/SIGMA
002117     165 AOA1 = AAIR+AGND
002121         X = (AAIR*XGND+AGND*XAIR)/AOA1
```

```
002125          A = AAIR*AGND/AOA1
002127          GO TO 180
002127      170 IF (L.GT.LGNDP1) GO TO 180
002133          X = XGND
002134          A = AGND
002135      180 E(L,K)=X*EK(L) + A*(-XJHH(L) + (B(LP1,K)*STH(LP1) - B(L,K)*STH(L))
                1 /R/ECON(L))
002156          IF(ABS(E(L,K)).LT.1.E-25) E(L,K)=0.
002167      185 CONTINUE                                                    TH ELOOP
C
C
C           THETA LOOP F
C
002172          DO 235 L=2,LMAX
002173          FK(L)=F(L,KP1)
002177          IF (L.GT.1) GO TO 200
002202          EVAL=.5*(EK(1)+E(1,KP1))
002206          GO TO 210
002207      200 IF (L.GT.LGND) GO TO 220
002213          LM1 = L-1
002214          EVAL=.5*(EFACTA(LM1)*(EK(L)+E(L,KP1))+EFACTB(LM1)*(EK(LM1)+E(LM1,K
               1P1)))
002231      210 SIG(L)=CEXN(L)*(EMOBFN(SQRT(EVAL**2+FK(L)**2)*EMOBCN))+CXXXPX(L)
002245          TEMP = SIG(L)*DTEPS
002247          IF (TEMP.LE.15.) GO TO 213
002251          X = 0.0
002252          A = 1.0/SIG(L)
002254          GO TO 230
002255      213 IF (TEMP.GE.1.E-5) GO TO 215
002260          X=1.-TEMP+.5*TEMP**2
002264          A=(1.-.5*TEMP)*DTEPS
002267          GO TO 230
002267      215 X=EXP(-TEMP)
002273          A = (1.0-X)/SIG(L)
002276          GO TO 230
002277      220 IF (L.GT.LGNDP1) GO TO 230
002303          X = XGND
002304          A = AGND
002305      230 F(L,KP1)=X*FK(L)-A*(RPDR*B(L,KP1)-R*B(L,K))/RPHDR/CMUDR-A*XJT(L)
002326          IF(ABS(F(L,KP1)).LT.1.E-25) F(L,KP1)=0.
002335      235 CONTINUE                                                    TH FLOOP
C
C           THETA LOOP B
C
002340          DO 245 L=2,LMAX
002341          LM1 = L-1
002343          B(L,K)=B(L,K)+DTOR*(E(L,K)-E(LM1,K))/DHTH(LM1)-DTORDR*(RPHDR*F(L,K
               1P1)-RMHDR*F(L,K))
002371          IF(ABS(B(L,K)).LT.1.E-25) B(L,K)=0.
002377      245 CONTINUE                                                    TH BLOOP
C
C           OUTPUT EDITING SECTION
C           KNUM = RANGE NO.
C           K = INDEX OF RANGE LOOP
C           KOUT(KNUM) = INDEX OF RANGE LOOP FOR RANGE KNUM
C           N = INDEX OF TIME LOOP
C           NBEGIN(KNUM) = INDEX OF TIME LOOP FOR RANGE KNUM
C
```

```
002402          DO 250 KNUM=1,NKOUT
002403          IF(K+NSHFTT+KDIM*LOADM1.EQ.KOUT(KNUM)) GO TO 260
002410       250 CONTINUE
002413          GO TO 370
002413       260 NVAL=N-NBEGIN(KNUM)
002416          IF (NVAL.LT.NSTART) GO TO 370
002420          DO 270 NN=1,NSTEPS
002422          IF (NVAL.LE.NSTOP(NN)) GO TO 280
002425       270 CONTINUE
002427          GO TO 370
002430       280 IF (MOD(NSTOP(NN)-NVAL,NSTEP(NN)).NE.0) GO TO 370
002436          CALL PRINTER
002437       370 CONTINUE                                                    R   LOOP
002442          CALL SEGMENT(2)
002443          GO TO 120
002444       380 IF(LOAD.NE.0) CALL SEGMENT(3)
002447       410 CONTINUE                                                    T   LOOP
            C
            C        FOR CK RUNS IRENUM IS NEG, PRINT ONLY, NO TAPES
            C
002452       420 IF(IRENUM.LT.0) GO TO 510
002454          IF(NLOAD.GT.1) GO TO 430
002457          CALL SEGMENT(4)
002460          GO TO 435
002461       430 JSTORE=ISTORE
002463       435 END FILE JDATA
            C
            C        WRITE LAST RESTART RECORD ON LDATA
            C
002465          NBLOCK=NBLOCK+1
002467          REWIND LDATA
002471          NMIN = NMAX + 1
002473          IF(NTIME.LT.NMAX) NMIN=NTIME
002476          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NMIN,1,3)
002514          DO 505 N=1,NLOAD
002515          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(1,2),XNM(1
               1 ,2),NLOAD,NMIN,KDIMM1,4)
002543          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1)
               1 ,XNE(1,KDIMP1),XNM(1,KDIMP1),NLOAD,NMIN,1,4)
002575       505 CONTINUE
            C
002600          END FILE LDATA
            C
            C        COMPLETE PRINTED OUTPUT
            C
002602       510 PRINT 610,HEADER
002610          PRINT 611
002614          DO 601 I=1,NFILES
002616          PRINT 612,I
002623          DO 601 K=1,NKOUT
002625          PRINT 613, K,ORMG(K),ICOUNT(K,I)
002640       601 CONTINUE
            C
002645          PRINT 24, NBLOCK
002653          IF(IRENUM.GE.0) T=NMIN*DT
002660          PRINT 614,T
002666          PRINT 615
002672          STOP
002674          END
```

```
                FUNCTION EMUBFN(EOVERP)
C
000003          IF (EOVERP.GT.0.4577) GO TO 3
000006          IF(EOVERP.GT.0.0905) GO TO 2
000011          IF(EOVERP.GT.0.0303) GO TO 1
000014          EMUBFN= 2.092/(1.0+33.0*EOVERP)
000020          RETURN
000020        1 EMUBFN=1.00922+EOVERP*(-34.1606+EOVERP*(332.685-EOVERP*1186.94))
000027          RETURN
000027        2 EMUBFN=0.169267/SQRT(EOVERP)
000033          RETURN
000033        3 IF(EOVERP.GT.60.0) GO TO 4
000037          XLN=ALOG(EOVERP)
000041          EMUBFN=0.16212*EXP(ALN*(0.64912*XLN-0.51686))
000051          RETURN
000051        4 EMUBFN=0.033684+0.6579/EOVERP
000055          RETURN
000056          END
```

```
      SUBROUTINE XJAY(T,R,X,KX,XJRH,XJT,S)
000012 COMMON       E(23, 251),   F(24, 251),   B(24, 251),  XNE(24,251),
     1            XNM(24,251),     CEX(24),    CA2XPX(24),  CXXAPX(24),
     2              CEXN(24),     KOUT(16),   KPRINT(16),ICOUNT(16,10),
     3             NBEGIN(16),    DMTH(23),   EFACTA(23),   EFACTB(23),
     4               HTH(23),     ECON(23),       EK(23),       SIG(24),
     5             FFACTA(23),   FFACTB(23),       TH(24),       STH(24),
     6                FK(24),    HEADER(8),     HEAD(8), OUTARR(25,9),
     7              FTEMP(24),   FTEMP2(24),   FATRAP(24),     THETA(24),
     8             HTHETA(23)
000012 COMMON /PARAMS/       C,            Q,          CMU,          EPS,
     1                  EPSOND,       SOND,         RHO,      ATTACH,
     2                   XELEN,      XIOMUB,      XELRGE,      XIOCOM,
     3                    ALF,        OCON,         NTH,        NHTH,
     4                   SCALE
000012 COMMON/SOURCE/      AJ1,         XJ2,         XJ3,         XJ4,
     1                     T0,          T1,          T2,          T3,
     2                   XJRON,      CLAMDA
000012       DIMENSION XJRH(1),XJT(1),S(1)
000012       CX=X=LRGE*Q/CLAMDA
000014       CXA=QCON*CX
000016       GDOT=TIMDEP(T_R/C)
000026       GDOTX=TIMDEP(TX-RX/C)
000036       DO 30 I=1,NTH
000040       X=R*COS(TH(I))/SCALE
000052       IF(ABS(X).LT.1.0E-6) GO TO 10
000055       XX=EXP(-X)
000060       X=R*RHO*(1.0-XX)/X
000067       GO TO 20
000067  10   XX=1.0
000071       X=R*RHO
000073  20   CONTINUE
000073       S(I)=GDOT*EXP(-X/(1.293*CLAMDA))*CXA*XX/(12.56637*R**2)
000114       XJT(I)=0.0
000116  30   CONTINUE
000120       DO 60 I=1,NHTH
000121       X=RX*COS(HTH(I))/SCALE
000133       IF(ABS(X).LT.1.0E-6) GO TO 40
000136       XX=EXP(-X)
000141       X=RX*RHO*(1.0-XX)/X
000150       GO TO 50
000150  40   XX=1.0
000152       X=RX*RHO
000154  50   CONTINUE
000154       XJRH(I)=GDOTX*EXP(-X/(1.293*CLAMDA))*CX*XX/(12.56637*RX**2)
000174  60   CONTINUE
000177       RETURN
000177       END
```

```
                FUNCTION TIMDEP(TAU)
000003          COMMON/SOURCE/      XJ1,          XJ2,          XJ3,          XJ4,
                1                    T0,           T1,           T2,           T3,
                2                 XJRUN,        CLAMDA
                C
000003    10    IF(TAU.GE.T2) GO TO 20
000006          TIMDEP=-(7.368E26*EXP(2.E6*TAU))/(1.+1.816E-4*EXP(2.5E6*TAU))
000024          RETURN
000024    20    TIMDEP=-(2.807E28*EXP(-.003E8*TAU))
000032          RETURN
000032          END
```

```
      SUBROUTINE RESTART
000002 COMMON       E(23, 251),   F(24, 251),   B(24, 251),  XNE(24,251),
     1             XNM(24,251),     CEX(24),   CA2XPX(24),   CX*XPX(24),
     2               CEXN(24),    KOUT(16),   KPRINT(16),ICOUNT(16,10),
     3             NBEGIN(16),     DMTH(23),   EFACTA(23),   EFACTB(23),
     4                HTH(23),     ECON(23),      EK(23),      SIG(24),
     5             FFACTA(23),   FFACTB(23),      TH(24),      STH(24),
     6                FK(24),    HEADER(8),    HEAD(8), OUTARR(25,9),
     7              FTEMP(24),   FTEMP2(24),   FATHAP(24),    THETA(24),
     8             HTHETA(23)
C
000002  COMMON /INTEGER/     KDATA,     ISTORE,      NTIME,      LOAD,
     1            NLOAD,      LMAX,    LMAXP1,       KDIM,    KDIMP1,
     2           IRENUM,     NKOUT,     LDATA,    NBLOCK,     JDATA,
     3           JSTORE,    NSHIFT,    KDIMM1,    LMAXp2,     NPACK,
     4             IREM,     IDATA
C
000002  COMMON /RANDT/           R,       ROC,       RPDR,      RPHDR,
     1           RPHDRC,    KSTART,        DR,        HDR,       LGM1,
     2             LGND,    LGNDP1,         K,        Kp1,       KNUM,
     3           NFILES,    NSHFTT,                 EMOCON,    EMOBCN,
     4            CMUDR,     DIEPS,     XTHAP,          T,       TPDT,
     5            TPHDT,     TMHDT,        DT,       AGND,      XGND,
     6         ORNG(16),   XJRM(24),    XJT(24),     Q(24)
C
000002  COMMON /SAVE/     ES(23,20), FS(24,20), BS(24,20),XNES(24,20),
     1    XNMS(24,20)
C
000002    1 FORMAT(////10H NBLOCK = ,I5)
000002      NBLOCK=NBLOCK+1
000004      REWIND LDATA
000006      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NTIME,1,5)
000024      DO 400 I=1,NLOAD
000026      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(1,2),XNM(1
     1 ,2),NLOAD,NTIME,KDIMM1,6)
000053      CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1)
     1 ,XNE(1,KDIMP1),XNM(1,KDIMP1),NLOAD,NTIME,1,6)
000105    400 CONTINUE
000110      END FILE LDATA
000112      REWIND ISTORE
000114      PRINT 1, NBLOCK
000122      RETURN
000123      END
```

```
                SUBROUTINE SEGMENT(ISEC1)
000003          COMMON       E(23, 251),   F(24, 251),   B(24, 251),   XNE(24,251),
                1           XNM(24,251),     CEX(24),   CA2XPX(24),   CXXXPX(24),
                2             CEXN(24),    KOUT(16),  KPRINT(16),ICOUNT(16,10),
                3            NBEGIN(16),    UMTM(23),   EFACTA(23),   EFACTB(23),
                4               HTM(23),    ECON(23),      EK(23),      SIG(24),
                5            FFACTA(23),   FFACTB(23),      TM(24),      STM(24),
                6                FR(24),    HEADER(8),    HEAD(8), OUTARR(25,9),
                7             FTEMP(24),   FTEMP2(24),   FATRAP(24),    THETA(24),
                8            HTHETA(23)
                C
000003          COMMON /INTEGER/     KUATA,     ISTORE,      NTIME,       LOAD,
                1           NLOAD,      LMAX,    LMAXP1,       KJIM,    KDIMP1,
                2          IRENUM,     NKOUT,     LDATA,     NBLOCK,      JDATA,
                3          JSTORE,    NSHIFT,    KDIMM1,    LMAXP2,      NPACK,
                4            IREM,     IDATA
                C
000003          COMMON /SAVE/    ES(23,20), FS(24,20), BS(24,20),XNES(24,20),
                1    XNMS(24,20)
                C
000003          GO TO (100,200,300,400), ISECT
                C
                C       READ IN FROM DISK LEFT MOST SEG TO START NEW LEFT MOST SEG AT
                C       NEW DELTA T
                C
000013        100 CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NTIME,1,7)
000031          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(1,2),XNM(1
                1 ,2),NLOAD,NTIME,KDIMM1,7)
000060          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1)
                1 ,XNE(1,KDIMP1),XNM(1,KDIMP1),NLOAD,NTIME,1,7)
000113          RETURN
                C
000114        200 IF (LOAD.EQ.0) LOAD=1
                C
                C       SAVE SEGMENT JUST COMPUTED ON DISK TO FREE MEMORY
                C
000117          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NTIME,1,8)
000135          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(1,2),XNM(1
                1 ,2),NLOAD,NTIME,KDIMM1,8)
                C
000164          IF(LOAD.EQ.1) GO TO 220
000167          LOADM1=LOAD-1
000170          DO 21, L=1,LMAX
000172          ES(L,LOADM1)=E(L,2)
000176          FS(L,LOADM1)=F(L,2)
000202          BS(L,LOADM1)=B(L,2)
000205          XNES(L,LOADM1)=XNE(L,2)
000211          XNMS(L,LOADM1)=XNM(L,2)
000214        210 CONTINUE
000216          FS(LMAXP1,LOADM1)=F(LMAAP1,2)
000222          BS(LMAXP1,LOADM1)=B(LMAAP1,2)
000226          XNES(LMAXP1,LOADM1)=XNE(LMAXP1,2)
000231          XNMS(LMAXP1,LOADM1)=XNM(LMAXP1,2)
                C
                C       LOAD IS INCREMENTED WHEN THE RANGE LOOP TERMINATES
                C       NORMALLY. LOAD = THE NUMBER OF SEGMENTS ON DISK.
                C
```

```
000235      220 LOAD=LOAD+1
000237          DO 23. L=1,LMAX
000240          XNE(L,1)=XNE(L,KDIMP1)
000244          XNM(L,1)=XNM(L,KDIMP1)
000247          E(L,1)=E(L,KDIMP1)
000252          F(L,1)=F(L,KDIMP1)
000255          B(L,1)=B(L,KDIMP1)
000260      230 CONTINUE
000262          XNE(LMAXP1,1)=XNE(LMAXP1,KDIMP1)
000266          XNM(LMAXP1,1)=XNM(LMAXP1,KDIMP1)
000271          F(LMAXP1,1)=F(LMAXP1,KDIMP1)
000274          B(LMAXP1,1)=B(LMAXP1,KDIMP1)
000277          IF (LOAD.LE.NLOAD) GO TO 250
                C
                C       NLOAD IS ADVANCED WHEN A NEW COMPLETE SEGMENT W/ KDIM MEMBERS
                C       IS ADDED TO THE SEGMENTS BEING HELD ON DISK. NLOAD IS = TO
                C       NUMBER OF SEGMENTS ON DISK + 1
                C
000302          NLOAD=NLOAD+1
000303          DO 24. K=2,KDIMP1
000305          XNE(LMAXP1,K)=0.
000310          XNM(LMAXP1,K)=0.
000313          F(LMAXP1,K)=0.
000316          B(LMAXP1,K)=0.
000320          DO 24. L=1,LMAX
000322          XNE(L,K)=0.
000325          XNM(L,K)=0.
000330          E(L,K)=0.
000333          F(L,K)=0.
000335      240 B(L,K)=0.
000344          RETURN
000345      250 IF(NSHIFT.NE.2) GO TO 290
                C
                C       READ IN FROM DISK NEXT SEG TO COMP SAME SEG AT NEW DELTA T
                C
000346          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(1,2),XNM(1
                1 ,2),NLOAD,NTIME,KDIMM1,7)
000375          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1
                1 ),XNE(1,KDIMP1),XNM(1,KDIMP1),NLOAD,NTIME,1,7)
000430          RETURN
                C
000431      260 KDIMM2=KDIM-2
000433          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(1,2),XNM(1
                1 ,2),NLOAD,NTIME,KDIMM2,9)
000462          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIM),F(1,KDIM),B(1,KDIM),XNE(1
                1 ,KDIM),XNM(1,KDIM),NLOAD,NTIME,1,7)
000515          IF(LOAD.EQ.NLOAD) GO TO 275
000520          DO 27. L=1,LMAX
000522          E(L,KDIMP1)=ES(L,LOAD)
000531          F(L,KDIMP1)=FS(L,LOAD)
000536          B(L,KDIMP1)=BS(L,LOAD)
000544          XNE(L,KDIMP1)=XNES(L,LOAD)
000551          XNM(L,KDIMP1)=XNMS(L,LOAD)
000557      270 CONTINUE
000561          F(LMAXP1,KDIMP1)=FS(LMAAP1,LOAD)
000567          B(LMAXP1,KDIMP1)=BS(LMAAP1,LOAD)
000574          XNE(LMAXP1,KDIMP1)=XNES(LMAXP1,LOAD)
000602          XNM(LMAXP1,KDIMP1)=XNMS(LMAXP1,LOAD)
```

```
000607          RETURN
000610      275 DO 280 L=1,LMAX
000612          E(L,KDIMP1)=0.0
000616          F(L,KDIMP1)=0.0
000620          B(L,KDIMP1)=0.0
000623          XNE(L,KDIMP1)=0.0
000625          XNM(L,KDIMP1)=0.0
000630      280 CONTINUE
000632          F(LMAXP1,KDIMP1)=0.0
000635          B(LMAXP1,KDIMP1)=0.0
000640          XNE(LMAXP1,KDIMP1)=0.0
000642          XNM(LMAXP1,KDIMP1)=0.0
000645          RETURN
          C
          C         SAVE RIGHT MOST SEG ON DISK, PREPARE TO STEP TIME AND READ IN
          C         EARLIER COMPUTED SEG AT START OF TIME LOOP
          C
000645      300 CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NTIME,1,8)
000663          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(1,2),XNM(1
                1 ,2),NLOAD,NTIME,KDIMP1,8)
000712          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1)
                1 ,XNE(1,KDIMP1),XNM(1,KDIMP1),NLOAD,NTIME,1,8)
000745          REWIND ISTORE
000747          REWIND JSTORE
000751          I=ISTORE
000753          ISTORE=JSTORE
000754          JSTORE=I
000755          IF(LOAD.EQ.1) RETURN
000760          LOADM1=LOAD-1
000762          DO 350 L=1,LMAX
000764          ES(L,LOADM1)=E(L,2)
000770          FS(L,LOADM1)=F(L,2)
000774          BS(L,LOADM1)=B(L,2)
000777          XNES(L,LOADM1)=XNE(L,2)
001003          XNMS(L,LOADM1)=XNM(L,2)
001006      350 CONTINUE
001010          FS(LMAXP1,LOADM1)=F(LMAXP1,2)
001014          BS(LMAXP1,LOADM1)=B(LMAXP1,2)
001020          XNES(LMAXP1,LOADM1)=XNE(LMAXP1,2)
001023          XNMS(LMAXP1,LOADM1)=XNM(LMAXP1,2)
001027          LOAD=1
001030          RETURN
001030      400 CALL IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,NLOAD,NTIME,1,8)
001046          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,2),F(1,2),B(1,2),XNE(1,2),XNM(1
                1 ,2),NLOAD,NTIME,KDIMP1,8)
001075          CALL IOSAVE(XNES,XNMS,ES,FS,BS,E(1,KDIMP1),F(1,KDIMP1),B(1,KDIMP1)
                1 ,XNE(1,KDIMP1),XNM(1,KDIMP1),NLOAD,NTIME,1,8)
001130          END FILE JSTORE
001132          REWIND JSTORE
001134          RETURN
001135          END
```

```
      SUBROUTINE PRINTER
000002      COMMON       E(23, 251),   F(24, 251),   B(24, 251),   XNE(24,251),
     1             XNM(24,251),     CEX(24),    CA2XPX(24),    CXXXPX(24),
     2               CEXN(24),    KOUT(16),  KPRINT(16),ICOUNT(16,10),
     3             NBEGIN(16),    DHTH(23),   EFACTA(23),   EFACTB(23),
     4                HTH(23),    ECON(23),      EK(23),       SIG(24),
     5             FFACTA(23),  FFACTB(23),      TH(24),       STH(24),
     6                FK(24),    HEADER(8),    HEAD(8), OUTARR(25,9),
     7              FTEMP(24),   FTEMP2(24),  FATRAP(24),     THETA(24),
     8             HTHETA(23)
C
C        **NOTE ** WHEN DIMENSIONS OF OUTARR ARE CHANGED,IROW CHECK
C        SHOULD BE CHANGED
C
000002      COMMON /INTEGER/      KDATA,     ISTORE,      NTIME,      LOAD,
     1            NLOAD,       LMAX,     LMAXP1,       KDIM,    KDIMP1,
     2            IRENUM,      NKOUT,      LDATA,    NBLOCK,     JDATA,
     3            JSTORE,     NSHIFT,    KDIMM1,    LMAXP2,     NPACK,
     4              IREM,      IDATA
C
000002      COMMON /RANDT/           R,        ROC,       RPDR,      RPHDR,
     1           RPHDRC,     KSTART,        DR,        HDR,       LGM1,
     2             LGND,      LGNDP1,         K,        KP1,       KNUM,
     3            NFILES,     NSHIFT,                  EMOCON,    EMOBCN,
     4             CMUDR,      DTEPS,     XTRAP,          T,       TPDT,
     5             TPHDT,      TMHDT,        DT,       AGND,       XGND,
     6          ORNG(16),   XJRM(24),   XJT(24),      Q(24)
C
000002    1 FORMAT(///3110)
000002    9 FORMAT(1H1/////1X,8A1,,18X,*RANGE NO. *,I2,11X,*PAGE *,I3)
000002   10 FORMAT(//11X,*TIME NION,NSEC*,6X,*TIME TOT SIGMA*,6X,*TIME B PHI*,
     1 10X,*TIME E THETA*,17X,*IRENUM = *,13,5X,*TIME E RADIAL*)
000002   11 FORMAT(/5X,4(5X,E15.8),31X,E15.8)
000002   12 FORMAT(/3X,*NION = *,E15.8,15X,*CURRENT = *,E15.8)
000002   13 FORMAT (/2X,*THETA*,6X,*NSEC*,9X,*NION*,7X,*XJR*,9X,*XJT*,10X,*Q*,
     17X,*SIGMA TOTAL*,3X,*B PHI*,7X,*E THETA*,6X,*THETA/2*,3X,
     2 *E RADIAL*/)
000002   14 FORMAT (F6.3,2E12.4,12X,5E12.4)
000002   15 FORMAT (32X,E12.4,64X,F6.3,E12.4)
000002   19 FORMAT(F8.3,17X,3(5X,E15.8))
000002   22 FORMAT (2X,*GROUND*,24(1H-),E12.4,2X,27(1H-),*GROUND*,1X,2E12.4,
     1 3X,F8.3,E12.4)
000002   23 FORMAT(F8.3,2E12.4,12X,2E12.4,12X,2E12.4)
C
000002      OUTARR(1,1)=T
000004      OUTARR(1,2)=T-RPHDRC
000006      OUTARR(1,3)=OUTARR(I,2)
000007      OUTARR(1,4)=TPDT-ROC
000011      OUTARR(1,5)=TPHDT-RPHDRC
000012      OUTARR(1,7)=TPHDT-ROC
000014      OUTARR(2,1)=ORNG(KNUM)
000016      IF(LGND.GT.LMAXP1) GO TO 200
000021      OUTARR(1,8)=B(LGM1,K)+XIRAP*(B(LGND,K)-B(LGM1,K))
000031      OUTARR(1,9)=F(LGM1,K)+XIRAP*(F(LGND,K)-F(LGM1,K))
000037  200 DO 310 L=1,LMAXP1
000041      LP1=L+1
000043      LP2=L+2
```

```
000045          IF (L.GT.LGND) GO TO 300
000050          OUTARR(LP1,3)=SIG(L)
000052      300 OUTARR(LP1,4)=B(L,K)
000057          OUTARR(LP1,5)=F(L,K)
000062          OUTARR(LP1,6)=Q(L)
000063          OUTARR(LP1,8)=XNM(L,KP1)
000067          OUTARR(LP1,9)=XJT(L)
000070          OUTARR(LP1,2)=XNE(L,KP1)
000074          IF (L.EQ.LMAXP1) GO TO 310
000076          OUTARR(LP1,7)=E(L,K)
000102          OUTARR(LP2,1)=XJRH(L)
000104      310 CONTINUE
000107          ICOUNT(KNUM,NFILES)=ICOUNT(KNUM,NFILES)+1
      C
000112          IF(KPRINT(KNUM).EQ.0) GO TO 380
000113          KPRINT(KNUM)=KPRINT(KNUM)-1
000115          IPAGE = 0
000115          DO 314 J=1,NFILES
000117          IPAGE = IPAGE + ICOUNT(KNUM,J)
000123      314 CONTINUE
000125          PRINT  9,HEADER,KNUM,IPAGE
000137          PRINT 10,IRENUM
000145          PRINT 11,(OUTARR(1,J),J=2,5),OUTARR(1,7)
000162          PRINT 13
000166          PRINT 14, THETA(1),OUTARR(2,2),OUTARR(2,8),OUTARR(2,9),OUTARR(2,6)
               1 ,(OUTARR(2,J),J=3,5)
000213          LIMIT=LGM1
000215          IF(LGND.GT.LMAX) LIMIT=LMAX
000220          DO 320 M=1,LIMIT
000222          L=M+2
000224          LMO=M+1
000226          PRINT 15, OUTARR(L,1),HTHETA(M),OUTARR(LMO,7)
000237          PRINT 14, THETA(LMO),OUTARR(L,2),OUTARR(L,8),OUTARR(L,9),OUTARR(L,
               1 6),(OUTARR(L,J),J=3,5)
000265      320 CONTINUE
000270          IF(LGND.GT.LMAX) GO TO 370
000273          PRINT 22, OUTARR(L+1,1),OUTARR(1,8),OUTARR(1,9),HTHETA(LMO),OUTARR
               1 (L,7)
000310          IF(LGND.EQ.LMAX) GO TO 370
000312          LMMO=LMAX-1
000314          DO 330 M=LGND,LMMO
000315          L=M+1
000317          LPJ=M+2
000321          MPJ=M+3
000323          PRINT 23, THETA(L),OUTARR(LP3,2),OUTARR(LP3,8),OUTARR(LP3,9),
               1 OUTARR(LP3,6),(OUTARR(LP3,J),J=4,5)
000350          PRINT 15, OUTARR(MP3,1),HTHETA(L),OUTARR(LP3,7)
000362      330 CONTINUE
000365      340 LST=LMAX+2
000367          LST1=LMAX+1
000371          PRINT 23, THETA(LST1),OUTARR(LST,2),OUTARR(LST,8),OUTARR(LST,9),
               1 OUTARR(LST,6),(OUTARR(LST,J),J=4,5)
000416      370 PRINT 1, NSHIFT,NSHIFT,NTIME
000430      380 IF(IRENUM.LT.0) RETURN
      C
      C         WRITE OUTPUT DATA ON TAPE JDATA
000432          IROW=3
000433          ICOL=1
```

```
000434          I=3
000435          II=4
000436          J=1
000437          JJ=1
000440          DO 400 IPACK=1,NPACK
000442          OUTARR(IROW,ICOL)=PACK(OUTARR(I,J),OUTARR(II,JJ))
000457          IROW=IROW+1
000460          IF (IROW.LE.25) GO TO 385
000462          IROW=1
000463          ICOL=ICOL+1
000464      385 I=I+2
000466          II=II+2
000467          IF(I.LT.LMAXP2) GO TO 400
000471          ITEST=I-LMAXP1
000473          GO TO (388,390,392), ITEST
000501      388 II=1
000502          JJ=JJ+1
000504          GO TO 400
000504      390 I=1
000505          J=J+1
000507          II=I+1
000510          JJ=JJ+1
000511          GO TO 400
000512      392 I=2
000513          J=J+1
000515      400 CONTINUE
000520          NPACKS=NPACK+2
000522          IF(IREM.EQ.0) GO TO 410
000523          OUTARR(IROW,ICOL)=OUTARR(LMAXP2,9)
000530          NPACKS=NPACKS+1
000531      410 CALL DATAPE(OUTARR,NPACKS,2)
000534          RETURN
000535          END
```

```
      SUBROUTINE SHIFT
000002      COMMON       E(23, 251),   F(24, 251),   B(24, 251),  XNE(24,251),
     1           XNM(24,251),      CEX(24),   CA2XPX(24),   CXXPX(24),
     2             CEXN(24),     KOUT(16),   KPRINT(16),ICOUNT(16,10),
     3            NBEGIN(16),     DMTH(23),   EFACTA(23),   EFACTB(23),
     4               HTH(23),     ECON(23),       FK(23),      SIG(24),
     5           FFACTA(23),   FFACTB(23),       TH(24),      STH(24),
     6               FK(24),    HEADER(8),     HEAD(8), OUTARR(25,9),
     7            FTEMP(24),   FTEMP2(24),   FXTRAP(24),    THETA(24),
     8           HTHETA(23)
      C
000002      COMMON /PARAMS/          C,     COULOM,       CMU,       EPS,
     1          EPSGND,        SGND,        RMU,     ATTACH,      XELEN,
     2          XIOMOB,     XELRGE,     XIOCOM,        ALF,       QCON,
     3          ITHETA,     HTHETA,      SCALE
      C
000002      COMMON /INTEGER/      KDATA,     ISTORE,      NTIME,       LOAD,
     1           NLOAD,       LMAX,     LMAXP1,       KDIM,     KDIMP1,
     2          IRENUM,      NKOUT,      LDATA,     NBLOCK,      JDATA,
     3          JSTORE,     NSHIFT,     KDIMM1,     LMAXP2,      NPACK,
     4            IREM,      IDATA
      C
000002      COMMON /RANDT/            R,       RUC,       RPDR,      RPHDR,
     1          RPHDRC,     RSTART,        DR,       HDR,       LGM1,
     2            LGND,     LGNDP1,         K,        KP1,       KNUM,
     3          NFILES,     NSAVTT,                 EMUCON,    EMUBCN,
     4           CMUDR,      DTEPS,     XTRAP,          T,       TPDT,
     5            TPHDT,      THDT,        DT,      AGND,       XGND,
     6         ORNG(16),   XJRH(24),   XJT(24),      Q(24)
      C
000002      COMMON /SAVE/     ES(23,20), FS(24,20), BS(24,20),XNES(24,20),
     1     XNMS(24,20)
000002      R=RSTART+HDR
000004      RPHDR=R+HDR
000006      RPDR=R+DR
000010      CALL XJAY(TMHDT,RPHDR,T,R,XJRH,XJT,Q)
000016      DO 1115 L=1,LMAXP1
000020      XNEX=XNE(L,2)
000022      XNMX=XNM(L,2)
000023      TEMP=ATTACH + ALF*(XNEX+XNMX)
000027      ARG = DT*TEMP
000031      EXPON = EXP(-ARG)
000034      IF(ARG .GE. 1.E-7) GO TO 1111
000036      XNE(L,1)=(1.0-ARG)*XNEX+DT*Q(L)
000044      GO TO 1112
000044 1111 FACTOR = 1. - EXPON
000046      XNE(L,1)=EXPON*XNEX + FACTOR*Q(L)/TEMP
000053 1112 TEMP=XIOCOM*(XNEX+XNMX)
000056      ARG = DT*TEMP
000060      EXPON = EXP(-ARG)
000063      IF(ARG .GE. 1.E-7) GO TO 113
000065      XNM(L,1)=XNMX*(1.-ARG)+DT*ATTACH*XNEX
000074      GO TO 1115
000074  113 FACTOR = 1. - EXPON
000076  114 XNM(L,1)=EXPON*XNMX
000101      XNM(L,1)=XNM(L,1) + FACTOR*ATTACH*XNEX/TEMP
000106 1115 CONTINUE
```

```
000111          DO 1120 L=1,LMAXP1
000112          CEAN(L)=COULOM*EMOCON*XNE(L,1)
000115          CXXXPX(L)=COULOM*XIONOB*(2.*XNM(L,1) + XNE(L,1))
000122     1120 CONTINUE
000124          DO 235 L=2,LMAX
000126          FK(L)=F(L,2)
000130          IF (L.GT.1) GO TO 200
000133          EVAL=.5*(E(1,1)+E(1,2))
000136          GO TO 210
000136      200 IF (L.GT.LGND) GO TO 220
000142          LM1 = L-1
000143          EVAL=.5*(EFACTA(LM1)*(E(L,1)+E(L,2))+EFACTB(LM1)*(E(LM1,1)+E(LM1,2
                1 )))
000153      210 SIG(L)=CEAN(L)*(EMOBCN(SQRT(EVAL**2+FK(L)**2)*EMOBCN))+CXXXPX(L)
000167          TEMP = SIG(L)*DTEPS
000171          IF (TEMP.LE.15.) GO TO 213
000173          X = 0.0
000174          A = 1.0/SIG(L)
000176          GO TO 230
000177      213 IF (TEMP.GE.1.E-5) GO TO 215
000202          X=1.-TEMP+.5*TEMP**2
000206          A=(1.-.5*TEMP)*DTEPS
000211          GO TO 230
000211      215 X=EXP(-TEMP)
000215          A = (1.0-X)/SIG(L)
000220          GO TO 230
000221      220 IF (L.GT.LGNDP1) GO TO 230
000225          X = AGND
000226          A = AGND
000227      230 F(L,1)=X*FK(L)-A*(RPMDR*B(L,2)-R*B(L,1))/RPMDR/CMUDR-A*XJT(L)
000241          IF(ABS(F(L,1)).LT.1.E-25) F(L,1)=0.0
000246      235 CONTINUE
000251          B(LMAXP1,1)=B(LMAXP1,2)
000253          DO 650  L=1,LMAX
000254          E(L,1)=E(L,2)
000256          B(L,1)=B(L,2)
000257      650 CONTINUE
000261          DO 700  K=2,KDIM
000263          KP1=K+1
000265          F(LMAXP1,K)=F(LMAXP1,KP1)
000272          B(LMAXP1,K)=B(LMAXP1,KP1)
000277          XNE(LMAXP1,K)=XNE(LMAXP1,KP1)
000305          XNM(LMAXP1,K)=XNM(LMAXP1,KP1)
000312          DO 700  L=1,LMAX
000314          E(L,K)=E(L,KP1)
000323          F(L,K)=F(L,KP1)
000330          B(L,K)=B(L,KP1)
000336          XNE(L,K)=XNE(L,KP1)
000343          XNM(L,K)=XNM(L,KP1)
000351      700 CONTINUE
000355          IF(LOAD.GT.0) GO TO 800
000360          F(LMAXP1,KDIMP1)=0.
000363          B(LMAXP1,KDIMP1)=0.
000366          XNE(LMAXP1,KDIMP1)=0.0
000370          XNM(LMAXP1,KDIMP1)=0.0
000373          DO 750  L=1,LMAX
000374          E(L,KDIMP1)=0.
000400          F(L,KDIMP1)=0.
```

```
000402          B(L,KD1MP1)=0.
000405          XNE(L,KD1MP1)=0.0
000407          XNM(L,KD1MP1)=0.0
000412      750 CONTINUE
000414          RETURN
000414      800 XNE(LMAXP1,KD1MP1)=ANES(LMAXP1,LOAD)
000422          XNM(LMAXP1,KD1MP1)=ANMS(LMAXP1,LOAD)
000430          F(LMAXP1,KD1MP1)=FS(LMAAP1,LOAD)
000435          B(LMAXP1,KD1MP1)=BS(LMAAP1,LOAD)
000443          DO 850 L=1,LMAX
000444          XNE(L,KD1MP1)=ANES(L,LOAD)
000452          XNM(L,KD1MP1)=ANMS(L,LOAD)
000457          E(L,KD1MP1)=ES(L,LOAD)
000466          F(L,KD1MP1)=FS(L,LOAD)
000473          B(L,KD1MP1)=BS(L,LOAD)
000501      850 CONTINUE
000503          RETURN
000503          END
```

```
      SUBROUTINE IOSAVE(XNES,XNMS,ES,FS,BS,E,F,B,XNE,XNM,JSET,NMIN,KSET,
     1 IUSE)
      COMMON       EE(23, 251),  FF(24, 251),  BB(24, 251), XXNE(24,251),
     1           XXNM(24,251),       CEX(24),   CA2XPX(24),   CXXPX(24),
     2              CEXN(24),      KOUT(16),   KMRINT(16),ICOUNT(16,10),
     3            NBEGIN(16),      DHTH(23),    EFACTA(23),   EFACTB(23),
     4               HTH(23),      ECON(23),       EK(23),       SIG(24),
     5            FFACTA(23),   FFACTB(23),       TH(24),       STH(24),
     6                FX(24),    HEADER(8),     HEAD(8), OUTARR(25,9),
     7             FTEMP(24),   FTEMP2(24),   FXTRAP(24),     THETA(24),
     8            HTHETA(23)
C
      COMMON /INTEGER/     KDATA,     ISTORE,      NTIME,      LOAD,
     1           NLOAD,      LMAX,     LMAXP1,      KDIM,    KDIMP1,
     2          IRENUM,     NKOUT,      LDATA,    NBLOCK,     JDATA,
     3          JSTORE,    NSHIFT,     KDIMM1,    LMAXP2,     NPACK,
     4            IREM,     IDATA
C
      COMMON /RANDT/            R,       RUC,       RPDR,     RPHDR,
     1          RPHDRC,    RSTART,        DR,        HDR,      LGM1,
     2            LGND,    LGNDP1,         K,        KP1,      KNUM,
     3          NFILES,    NSHFTT,                 EMOCON,   EMOBCN,
     4           CMUDR,     DTEPS,     XTRAP,          T,      TPDT,
     5           TPHDT,     THHDT,        DT,       AGND,     XGND,
     6        ORNG(16),   XJRM(24),   XJT(24),     Q(24)
C
      DIMENSION XNES(24,JSET),XNMS(24,JSET),ES(24,JSET),FS(24,JSET),BS
     1 (24,JSET),E(23,KSET),F(24,KSET),B(24,KSET),XNE(24,KSET),XNM(24,
     2 KSET),DUM1(23),DUM2(24)
      GO TO (100,200,300,400,500,600,700,800,900), IUSE
  100 READ(KDATA) ICOUNT,KMRINT,NMIN,LOAD,NLOAD,NBLOCK,NSHFTT,RSTART,
     1 FTEMP,FXTRAP,XNES,XNMS,ES,FS,BS
  200 READ (KDATA ) E,F,B,XNE,XNM
      WRITE(ISTORE) E,F,B,XNE,XNM
      RETURN
  300 WRITE(LDATA)ICOUNT,KMRINT,NMIN,LOAD,NLOAD,NBLOCK,NSHFTT,RSTART,
     1 FTEMP,FXTRAP,XNES,XNMS,ES,FS,BS
  400 READ (JSTORE) E,F,B,XNE,XNM
      WRITE(LDATA ) E,F,B,XNE,XNM
      RETURN
  500 WRITE(LDATA)ICOUNT,KMRINT,NMIN,LOAD,NLOAD,NBLOCK,NSHFTT,RSTART,
     1 FTEMP,FXTRAP,XNES,XNMS,ES,FS,BS
  600 IF(JSET.EQ.1) GO TO 650
      READ (ISTORE) E,F,B,XNE,XNM
  650 WRITE(LDATA ) E,F,B,XNE,XNM
      RETURN
  700 READ(ISTORE) E,F,B,XNE,XNM
      RETURN
  800 WRITE(JSTORE) E,F,B,XNE,XNM
      RETURN
  900 READ(ISTORE) DUM1,E,DUM2,F,DUM2,B,DUM3,XNE,DUM3,XNM
      RETURN
      END
```

```
                SUBROUTINE DATAPE(OUTARR,NPACKS,IOPT)
000006          COMMON /INTEGER/      KDATA,     ISTORE,     NTIME,      LOAD,
               1              NLOAD,      LMAX,    LMAAP1,     KDIM,    KDIMP1,
               2             IRENUM,    MAXOUT,     LDATA,   NBLOCK,     JDATA,
               3             JSTORE,    NSHIFT,    KDIMM1,   LMAXP2,     NPACK,
               4               IREM,     IDATA
000006          DIMENSION OUTARR(NPACK)
C
000006          GO TO (100,200), IOPT
000013      100 READ(IDATA) OUTARR
000022          IF(EOF,IDATA) 250,200
000027      200 WRITE(JDATA) OUTARR
000036          RETURN
000037      250 IOPT=2
000042          RETURN
000042          END
```

```
                BLOCK DATA
C
C*** TEST PROBLEM                    ***
C
000002          COMMON /PARAMS/          C,      COULOM,       CMU,       EPS,
               1          EPSGND,      SGND,       RMO,    ATTACH,     XELEN,
               2          XIOMOB,    XELMGE,    XIOCOM,       ALF,      QCON,
               3          ITMETA,    IATMET,     SCALE
C
000002          DATA                     C,      COULOM,       CMU,       EPS,
               1          EPSGND,      SGND,                 ATTACH,     XELEN,
               2          XIOMOB,    XELMGE,    XIOCOM,       ALF/
               3                       3.E8,  1.6E-19, 12.566E-7,.88416E-11,
               4  14.14656E-11,       5.E-3,                  1.E8,       1.4,
               5          2.7E-4,      4.51,    2.3E-12,    4.5E-13/
000002          END
```

```
                                IDENT PACK                                      00000000
                                ENTRY PACK                                      00000010
                                ENTRY UNPACK                                    00000020
  0  2001 3136000000000006       VFD  42/0LPACK+18/2                           00000030
  1  0006 2000006 6000000       PACK  DATA 0                                    00000040
  2  5611                            SA1  B1                                    00000050
          5140000012 +                SA4  ZLOHF                              00000060
                11111                 MX1  X1*X4                              00000070
  3  5622                            SA2  B2                                    00000080
        20236                         LX2  36B                                00000090
             5130000014 +             SA3  ZUPHF                              00000100
  4  1142                            BX2  X2*X3                              00000110
        12612                         BX6  X1*X2                              00000120
             0400000001 +             EQ   PACK                               00000130
  5  0006 0000000000000       UNPACK  DATA 0                                  00000140
  6  5611                            SA1  B1                                    00000150
        10611                         BX6  X1                                 00000160
        10711                         BX7  X1                                 00000170
             21639                    AX6  36B                                00000180
  7  2036                            LX6  36B                                00000190
        56620                         SA6  B2                                    00000200
        20736                         LX7  36B                                00000210
             21739                    AX7  36B                                00000220
 10  2036                            LX7  36B                                00000230
        56739                         SA7  B3                                    00000235
             0400000005 +             EQ   UNPACK                             00000240
               11 +       ZUPHF     LIT  0000000007777777778                  00000250
               12 +       ZLOHF     LIT  7777777770000000000B                 00000260
 13                                  END                                     00000280
             45603       STORAGE USED          24 STATEMENTS      4 SYMBOLS
                         6400 ASSEMBLY       0.235 SECONDS       10 REFERENCES
```

C. REDACT

1. Description of Modifications

Program REDACT has been modified and is now a subroutine with a driver program DATAPK. This modification was made in order to process data tapes from the modified ELECTRA which have a packed OUTARR (see II. A.1.).

DATAPK opens and closes the plot files and calculates the number of words that are packed in OUTARR. In subroutine REDACT, the column and row indicators for data editing have been modified to correspond to the modifications made in ELECTRA. In order to process the data tape, the column and row indicators for the particular variable to be plotted are used to determine the word to be unpacked. The data array is unpacked by an assembly language routine (UNPACK).

2. Program Listing

```
      PROGRAM DATAPK (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE10=1021,
     1 TAPE11,TAPE18=1001)
C
000003      COMMON ORD(1003),TH(1003),IBUF(1,24),HEADER(8),THETA(20),
     1 HTHETA(20),ORDTH(20),ANGLE(18),HANGLE(17),LMAXP1,LMAXP4
C
000003      DIMENSION DATA(84)
C
000003    1 FORMAT(1H1///31X,7A10,A8///)
C
000003      REWIND 18
000005      REWIND 11
000007      READ(11) NFILES,NRNGES,LMAX
000020      READ(11) HEADER
000025      WRITE (6,1) HEADER
000033      READ (11) ANGLE,HANGLE
C           INITIALIZE PLOT ROUTINE
C
000042      CALL PLOTS(IBUF,1024,10)
000045      CALL PLOT (0.,-11.,-3)
000050      CALL PLOT (0.,1.25,-3)
C
C           COMPUTE POINTERS FOR THETA
C
000053      LMAXP1=LMAX+1
000055      LMAXP4=LMAX+4
000056      NPACK=LMAXP4*2+1
000060      CALL REDACT (DATA,NPACK)
C
C           CLOSE PLOT FILE
C
000062      CALL PLOT (0.,0.,999)
000065      STOP
000067      END
```

```
      SUBROUTINE REDACT (DATA,NPACK)
C
C        DATA EDITING KEY
C
C
C                                              VAY(1)      NIUN      SIGT
C                                              VAY(2)      NSEC      SIGT
C                                    THETA     VAY(3)      SIGT      SIGT
C                                    THETA     VAY(4)      BPHI      BTAU
C                                    THETA     VAY(5)      ETHE      FTAU
C                                              VAY(6)      CRNT      BTAU
C                                HALF THETA    VAY(7)      ERAD      ETAU
C
C
      COMMON ORD(1003),TH(1003),IBUF(1124),HEADER(8),THETA(20),
     1 HTHETA(20),ORDTH(20),ANGLE(18),HANGLE(17),LMAXP1,LMAXP4
C
      DIMENSION DATA(NPACK)
      DIMENSION VAY(8),TAB(7),HEADR(4),WORD(2)
C
      EQUIVALENCE (HEADR,HEADER)
C
      DATA VAY /4HNIUN,4HNSEC,4HSIGT,4HBPHI,4HETHE,4HCRNT,4HERAD,4HSTOP/
      DATA LINES,12HPR,LOGO,LOGT /4HLINE,4HPR ,4HLOGO,4HLOGT/
      DATA TAB /10H   NIUN   ,10H   NSEC   ,10HTOT. SIGMA,10H  B - PHI ,
     1          10H  E THETA ,10H  CURRENT ,10H E RADIAL /
      DATA JTHETA,JTIME/10HTHETA ONLY,10H TIME ONLY/
      DATA ANEW,ALAB/4HNEW ,10H   TIME   /
C
    1 FORMAT(I10,7F10.0/(8F10.0))
    2 FORMAT(2X,A4,4X,F5.0,I5,6X,A4,F4.0,F10.0,2A,A4,F4.0,2X,A4,2F5.0,
     1 A2)
    3 FORMAT(/1X,41HRANGE NUMBER IS READ 0 OR NEGATIVE, EXIT.)
    4 FORMAT(/1X,40HTHETA INPUT IS READ 0 OR NEGATIVE, EXIT.)
    5 FORMAT(/1X,75HSYMBOL NAME MISSING, MISSPELLED OR NOT RIGHT ADJUSTE
     1D IN INPUT FIELD, EXIT.)
    6 FORMAT(/1X,25HALL DATA PROCESSED, EXIT.)
    7 FORMAT(//1X,29HTHETA PLOT COMPLETED FOR ,A4)
    8 FORMAT(//1X,A5,12H  AT RANGE ,F6.1,9H   THETA ,I2//1X,A5,F6.1,
     1   E16.2,2X,A5,F6.1,A5,2F6.1)
    9 FORMAT(6E20.8)
   10 FORMAT(F10.0)
   12 FORMAT(A10)
   15 FORMAT(I7,33X,I7)
   18 FORMAT(27H NO DATA ON TAPE FOR RANGE ,F10.2,21H, SKIP TO NEXT RANG
     1E.)
   21 FORMAT(33H NO DATA SAVED ON DISK FOR RANGE ,F10.2)
   22 FORMAT (1X,4I10)
C
C
C
C        IF JTYPE = JTIME, ONLY TIME HISTORY PLOTS ARE MADE
C
      ISORT=0
      ICONS=LMAXP4/2-1
      IREM=MOD(LMAXP4,2)
      IF (IREM.EQ.0) ICONS=LMAXP4/2-2
```

```
000016          WRITE (6,22) ICOMS,LMAXM4,NPACK,IREM
000032          READ 1,* HDR
000040          READ(5,12) JTYPE
000046          IF(JTYPE.E.JTIME) GO TO 100
000052          READ(5,1) ITMETA,(IMETA(I),I=1,ITMETA)
000067          READ(5,1) IMTHET,(MIMETA(I),I=1,IMTHET)
000106          ITM=
000107      100 READ(5,2) SYM,RCARD
000117          IF(SYM.NE.ANEXR) GO TO 100
                C
                C       READ ALL DATA FROM TAPE AND WRITE PLOT DATA ON DISC OR DRUM
                C
                C
000123      101 RTEMP=0.
000124          NPTS=
000125          IF(ISVRT.EQ.1) BACKSPACE 11
000131          ISVRT=
000132      102 READ(11) DATA
000140          IF(EOF.11) 115,103
000144      103 READDATA(2)
000146          IF(R.EQ.RTEMP) GO TO 110
000150          IF(ISVRT.EQ.1) GO TO 110
000152          IF(R-RDR.GT.RCARD) GO TO 115
000156          IF(R-RDR.GE.RCARD) GO TO 110
000160          GO TO 102
000161      105 WRITE(6,16) RCARD
000167          GO TO 100
000171      110 WRITE(18) DATA
000177          NPTS=NPTS+1
000201          RTEMP=R
000202          ISVRT=1
000203          GO TO 102
000205      112 WRITE(18) DATA
000213          NPTS=NPTS+1
000215          GO TO 102
                C       READ PLOT PARAMETER CARD
                C
000217      115 READIND 16
000221          READ(5,2) SYM,RCARD,ITM,ILOGO,DEC,IS,ILOGT,DECT,ILINE,SAB,SOR,IPR
000255          IF(SYM.EQ.ANEXR) GO TO 101
                C
                C       CHECK INPUT, IF WRONG SET STANDARD VALUES
                C
000261          IF(DEC.LE.0.) DEC=4.
000263          IF(SAB.LE.0.) SAB=0.
000266          IF(SOR.LE.0.) SOR=0.5
000271          IF(IS.LE.0.) IS=1.0E-6
000274          IF(DECT.LE.0.0) DECT=2.
                C
                C       INITALIZE PLOT COORDINATES
                C
000277          SPACE=SAB+2.5
000301          XRNG=SAB-.7
000303          IF(RCARD.GE.1000.) XRNG=XRNG-0.1
000307          XTM=SAB-.28
000311          XMEAN=SAB-3.25
000313          YMEAN=SOR-.17
000315          ZTM=ITM
```

```
000317          IRC=RCARD
                C
                C       INITIALIZE INDICATORS AND COUNTERS
                C
000321          JJ=0
                C
                C       IF SYM = VAY(8) , JOB IS FINISHED
                C
000322          IF(SYM.NE.VAY(8)) GO TO 116
000324          WRITE(6,6)
000327          RETURN
000330        116 WRITE(6,4) SYM,RCARD,ITH,ILOGO,DEC,IS,ILOGI,IFCT,ILINE,SAB,SOR
                C
                C       DIAGNOSTICS FOR CARD INPUT
                C
000362          IF(IRC.GT.0) GO TO 117
000366          WRITE(6,3)
000371          GO TO 115
000373        117 IF(SYM.EQ.VAY(1).OR.SYM.EQ.VAY(2).OR.SYM.EQ.VAY(8)) GO TO 119
000406          IF(ITH.GT.0) GO TO 119
000411          WRITE(6,4)
000414          GO TO 115
                C
                C       DETERMINE ROW AND COLUMN INDICATORS FOR DATA EDITING
                C
000416        119 IRC=ITH/2+3
000421          IRO=ITH/2+4
000423          IRI=1
000424          DO 13  I=1,7
000426          IC=I
000427          YLAB=TAB(IC)
000431          IF(SYM.EQ.VAY(I)) GO TO 132
000433        130 CONTINUE
000435          WRITE(6,5)
000440          GO TO 115
000442        132 GO TO (140,170,160,145,165,150,155),IC
000455        140 IC=2
000456          IR=3
000457          IT=3
000460          GO TO 175
000461        145 IR=IRO
000463          IT=4
000464          IF(ITH.EQ.LMAXP1) IR=2
000467          GO TO 175
000470        150 IC=4
000471          IR=3
000472          IT=4
000473          GO TO 175
000474        155 IR=IRE
000476          IC=2
000477          IT=2
000500          GO TO 175
000500        160 IR=IRO
000502          IT=3
000503          GO TO 175
000503        165 IR=IRO
000505          IC=1
000506          IT=1
```

```
000507          IRT=J
000510          IF (ITH.NE.LMAXP1) GO TO 175
000512          IR=2
000513          IC=2
000514          GO TO 175
000514      170 IC=3
000515          IR=3
000516          IT=3
                C
                C       DETERMINE LOCATION OF TIME AND VARIABLE DATA
                C
000517      175 ISET1=1
000520          ISET2=1
000521          WRITE (6,22) ICONS,LMAXP4,NPACK,IREM
000535          IF (IREM.NE.0) GO TO 180
000540          LOCV=MOD(IR,2)
000544          IF (LOCV.EQ.0) ISET2=2
000546          IR=IR-1
000550          IRT=IRT-1
000551          GO TO 185
000551      180 LOCT=MOD(IRT+IT,2)
000556          LOCV=MOD(IR+IC,2)
000564          IF (LOCT.NE.0) ISET1=2
000566          IF (LOCV.NE.0) ISET2=2
                C
                C       DETERMINE WORD TO BE UNPACKED
                C
000570      185 IUPACT=(IRT+IT*LMAXP4)/2-ICONS
000575          IUPACV=(IR+IC*LMAXP4)/2-ICONS
                C
                C       EDIT THE DATA
                C
000601          DO 200 L=1,NPTS
000603          READ(18) DATA
000611          CALL UNPACK (DATA(IUPACT),WORD(1),WORD(2))
000617          T=WORD(ISET1)
000623          IF(T.LT.TS) T=TS
000626          IF(JJ.EQ.1000) GO TO 250
000630          JJ=JJ+1
000631          CALL UNPACK (DATA(IUPACV),WORD(1),WORD(2))
000636          ORD(JJ)=WORD(ISET2)
000642          TM(JJ)=T
000644          IF (JJ.LE.10) PRINT, DATA(2),TM(JJ),ORD(JJ)
000667          IF(IPR.NE.I2HPR) GO TO 200
000671          WRITE(6,15) JJ
000677          WRITE(6,9) ORD(JJ),TM(JJ)
000714      200 CONTINUE
                C
                C       IF JTYPE = JTHETA, ONLY MAX VALUE PLOTS ARE MADE
                C
000720      250 IF(JTYPE.EQ.JTHETA) GO TO 650
                C
000722          IPLOT=1
000723          IF(ILOGT.NE.LOGT) GO TO 300
000725          IF(ILOGO.NE.LOGO) IPLOT=3
000730          GO TO 400
000731      300 IPLOT=2
000732          IF(ILOGO.NE.LOGO) IPLOT=4
```

```
000735      400 CALL GRAPH(IPLOT,TM,ORD,XLAB,YLAB,HEADR,SAB,SOR,JECT,DEC,ILINE,IPR
                1 ,ITM,IPC,JJ)
000754      500 IF (JTYPE.EQ.JTIME) GO TO 115
            C
            C       SECTION FOR GENERATING MAX VALUE VS. THETA PLOTS
            C
000760      650 IF(SYM.NE.VAY(4).AND.SYM.NE.VAY(5).AND.SYM.NE.VAY(7)) GO TO 115
000773          ITP=ITP+1
000775          YMAX=ABS(ORD(1))
000777          DO 655 I=2,JJ
001000          IF(ABS(ORD(I)).GT.YMAX) YMAX=ABS(ORD(I))
001006      655 CONTINUE
001011          ORDTM(ITP)=YMAX
001013          IF(IC.EQ.7) GO TO 660
001015          IF(ITP.NE.ITHETA) GO TO 115
001017          CALL THEPLT(ORDTH,THETA,SOR,SAB,ITP,YLAB,RCARD,HEADR)
001026          GO TO 665
001030      660 IF(ITP.NE.IMTHET) GO TO 115
001032          CALL THEPLT(ORDTH,MTHETA,SOR,SAB,ITP,YLAB,MCARD,HEADR)
001042      665 CALL PLOT(SPACE,0.,-3)
001045          ITP=0
001046          WRITE(6,7) SYM
            C
            C       RETURN FOR MORE DATA
            C
001054          GO TO 115
001056          END
```

```
      SUBROUTINE THEPLT(ORD,ABSC,SOR,SAB,ITP,TLAB,ZRT,HEADR)
000013      DIMENSION ORD(1),ABSC(1),TLAB(1),HEADR(1)
000013      XRNG=SAB-.6
000014      XHEAD=SAB-3.25
000016      YHEAD=SOR-.17
000021      CALL SCALE(ORD,SOR,ITP,1)
000023      CALL SCALE(ABSC,SAB,ITP,1)
000031      CALL NUMBER(XRNG,.07,.1,ZRT,0.,-1)
000036      CALL SYMBOL(XHEAD,YHEAD,.1,HEADR,0.,36)
000043      CALL LINE(ABSC,ORD,ITP,1,1,1)
000053      CALL AXIS(0.,0.,TLAB,10,SOR,90.,ORD(ITP+1),ORD(ITP+2))
000072      CALL AXIS(0.,0.,5HTHETA,-5,SAB,0.,ABSC(ITP+1),ABSC(ITP+2))
000111      CALL PLOT(0.,SOR,3)
000117      ISAB=SAB
000124      ZSAB=ISAB
000126      IF(ZSAB.EQ.SAB) ISAB=SAB-1.
000133      DO 200 L=1,ISAB
000135      XL=L
000136      CALL SYMBOL(XL,SOR,.1,10,0.,-2)
000142  200 CONTINUE
000150      CALL PLOT(SAB,SOR,2)
000152      ISOR=SOR
000157      ZSOR=ISOR
000161      IF(ZSOR.EQ.SOR) ISOR=SOR-1.
000166      ZSOR=ISOR
000167      CALL SYMBOL(SAB,ZSOR,.1,10,90.,-2)
000173      IS=ISOR-1
000175      DO 300 K=1,IS
000202      YK=ZSOR-K
000205      CALL SYMBOL(SAB,YK,.1,10,90.,-2)
000211  300 CONTINUE
000217      CALL PLOT(SAB,0.,2)
000221      X90=(90.-ABSC(ITP+1))/ABSC(ITP+2)
000231      CALL PLOT(X90,0.,3)
000233      CALL PLOT(X90,SOR,2)
000241      RETURN
000242      END
```

```
                                IDENT PACK                                                      00000000
                                ENTRY PACK                                                      00000010
                                ENTRY UNPACK                                                    00000020
 0  20[illegible]                VFD   42/6LPACK,18/2                                          00000030
 1  00[illegible]        PACK    DATA  0                                                       00000040
 2  5611[illegible]              SA1   B1                                                      00000050
        514[illegible]+           SA4   ZLOMF                                                 00000060
                  1111[illegible] BX1   X1*X4                                                00000070
 3  5[illegible]                 SA2   B2                                                    00000080
          [illegible]236          LX6   36B                                                  00000090
              513[illegible]+     SA3   ZUPMF                                                00000100
 4  11[illegible]                BX2   X2*X3                                                00000110
          12612                   BX6   X1+X6                                               00000120
              0400000001+         EQ    PACK                                                00000130
 5  00[illegible]        UNPACK  DATA  0                                                       00000140
 6  5611[illegible]              SA1   B1                                                    00000150
          1[illegible]611         BX6   X1                                                  00000160
              10711               BX7   X1                                                  00000170
                  21639           AX6   36B                                                 00000180
 7  2[illegible]                 LA6   36B                                                   00000190
          56623                   SA6   B2                                                  00000200
              23736               LA7   36B                                                 00000210
                  21739           AX7   36B                                                 00000220
10  20[illegible]                LA7   36B                                                   00000230
          56730                   SA7   B3                                                  00000235
              0400000005+         EQ    UNPACK                                              00000240
                    11+   ZUPMF   LIT   0000000000777777777778                              00000250
                    12+   ZLOMF   LIT   777777777700000000005                               00000260
13                              END                                                           00000280
                  45669   STORAGE USED          29 STATEMENTS      4 SYMBOLS
                          6400 ASSEMBLY       0.246 SECONDS       10 REFERENCES
```

D. REDACTO

1. Description of Modifications

Program REDACTO has been modified and is now a subroutine with a driver program DATAPKO. This modification was made in order to process data tapes from the modified ORESTES which have a packed OUTARR (see II.B.1).

DATAPKO opens and closes the plot files and calculates the number of words that are packed in OUTARR. In subroutine REDACTO, the column and row indicators for data editing have been modified to correspond to the modifications made in ORESTES. In order to process the data tape, the column and row indicators for the particular variable to be plotted are used to determine the word to be unpacked. The data array is unpacked by an assembly language routine (UNPACK).

2. Program Listing

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

              PROGRAM DATAPKU(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE10=1021,
             1 TAPE11,TAPE18=1001)
        C
000003        COMMON ORO(1003),TH(1003),IBUF(1024),HEADER(8),THETA(20),
             1  HTHETA(20),ORDTH(20),ANGLE(24),HANGLE(23),LMAX,LMAXP1,
             2  LMAXP2
        C
000003        DIMENSION DATA(225)
        C
000003       1 FORMAT(1H1///31X,7A10,A0///)
000003       2 FORMAT (1X,4I10)
        C
000003        REWIND 10
000005        REWIND 11
000007        READ(11) NFILES,NRNGES,LMAX
000020        READ(11) HEADER,IRCNUM
000027        READ (11) ANGLE,HANGLE
000036        WRITE (6,1) HEADER
        C
        C        COMPUTE POINTERS FOR THETA
        C
000044        LMAXP1=LMAX+1
000046        LMAXP2=LMAX+2
        C
        C        INITIALIZE PLOT ROUTINE
        C
000050        CALL PLOTS(IBUF,1024,10)
000052        CALL PLOT (0.,-11.,-3)
000055        CALL PLOT (0.,1.26,-3)
        C
000060        NPACK=(LMAXP2*9)/2+1
000063        IREM=MOD(LMAXP2*9,2)
000070        IF (IREM.NE.0) NPACK=NPACK+1
000073        WRITE (6,2) LMAXP1,LMAXP2,NPACK,IREM
000107        CALL REDACTO(DATA,NPACK)
        C
        C        CLOSE PLOT FILE
        C
000111        CALL PLOT (0.,0.,999)
000114        STOP
000116        END
```

```
      SUBROUTINE REDACTO (DATA,NPACK)
C
000005      COMMON ORD(1003),TM(1:03),IBUF(1;24),HEADER(5),THETA(20),
     1  RTHETA(20),ORDTH(20),ANGLE(24),RANGLE(23),LMAX,LMAXP1,
     2  LMAXP2
C
C        DATA EDITING KEY
C
C
C                                              VAY(1)      NION      SIGT
C                                              VAY(2)      NSEC      SIGT
C                                    THETA     VAY(3)      SIGT      SIGT
C                                    THETA     VAY(4)      BPHI      BTAU
C                                    THETA     VAY(5)      ETHE      FTAU
C                              HALF  THETA     VAY(6)      XJRH      ETAU
C                              HALF  THETA     VAY(7)      ERAD      ZTAU
C                                              VAY(8)      XJT       FTAU
C
000005      DIMENSION DATA(NPACK)
000005      DIMENSION VAY(9),TAB(5),HEADR(4) ,WORD(2)
C
000005      EQUIVALENCE (HEADR,HEADER)
C
000005      DATA VAY /4HNION,4HNSEC,4HSIGT,4HBPHI,4HETHE,4HXJRH,4HERAD,4H XJT,
     1 4HSTOP/
000005      DATA LINES,12HPR,LOGO,LOGT /4HLINE,2HPR,4HLOGO,4HLOGT/
000005      DATA TAB /10H   NION   ,10H   NSEC   ,10HTOT. SIGMA,10H  B - PHI ,
     1          10H  E THETA ,10H XJ RADIAL,10H E RADIAL ,10H    XJT   /
000005      DATA JTHETA,JTIME/1HTHETA ONLY,10H TIME ONLY/
000005      DATA AXES,XLAB/4HAXES,10H   TIME   /
C
000005     1 FORMAT(I10,7F10.0/(8F10.0))
000005     2 FORMAT(2X,A4,4X,F5.2,I3,6X,A4,F4.0,F10.0,2X,A4,F4.0,2X,A4,2F5.0,
     1 A2)
000005      3 FORMAT(/1X,41HRANGE NUMBER IS READ 0 OR NEGATIVE, EXIT.)
000005      4 FORMAT(/1X,40HTHETA INPUT IS READ 0 OR NEGATIVE, EXIT.)
000005      5 FORMAT(/1X,76HSYMBOL NAME MISSING, MISSPELLED OR NOT RIGHT ADJUSTE
     1D IN INPUT FIELD, EXIT.)
000005      6 FORMAT(/1X,25HALL DATA PROCESSED, EXIT.)
000005      7 FORMAT(//1X,26HTHETA PLOT COMPLETED FOR ,A4)
000005      8 FORMAT(//1X,A5,12H  AT RANGE ,F6.1,4H   THETA ,I2//1X,A5,F6.1,
     1   E1 .2,2X,A5,F6.1,A5,F6.1)
000005      9 FORMAT(6E20.8)
000005     10 FORMAT(F10.0,I10)
000005     12 FORMAT(A10)
000005     15 FORMAT(I7,33X,I7)
000005     18 FORMAT(27H NO DATA ON TAPE FOR RANGE ,F10.2,21H, SKIP TO NEXT RANG
     1E.)
000005     21 FORMAT(33H NO DATA SAVED ON DISK FOR RANGE ,F10.2)
000005     22 FORMAT (1X,4I10)
C
C
C
C        IF JTYPE = JTIME, ONLY TIME HISTORY PLOTS ARE MADE
C
000005      ISUBR=0
```

```
000006          ICONS=LMAXP2/2-1
000010          IREM=MOD(LMAXP2,2)
000014          IF (IREM.EQ.0) ICONS=LMAXP2/2-2
000016          WRITE (6,22) ICONS,LMAXP2,NPACK,IREM
000032          READ 10,HDR  ,LGND
000042          READ(5,12) JTYPE
000050          IF(JTYPE.EQ.JTIME) GO TO 100
000054          READ(5,1) ITHETA,(THETA(I),I=1,ITHETA)
000071          READ(5,1) IMTHET,(MTHETA(I),I=1,IMTHET)
000110          ITH=0
000111      100 READ(5,2) SYM,RCARD
000121          IF(SYM.NE.ANEWR) GO TO 100
                C
                C       READ ALL DATA FROM TAPE AND WRITE PLOT DATA ON DISC OR DRUM
                C
000125      101 RTEMP=0.
000126          NPTS=0
000127          IF(ISORT.EQ.1) BACKSPACE 11
000133          ISORT=0
000134      102 READ(11) DATA
000142          IF(EOF,11) 115,103
000146      103 R=DATA(2)
000150          IF(R.EQ.RTEMP) GO TO 110
000152          IF(ISORT.EQ.1) GO TO 112
000154          IF(R-HDR.GT.RCARD) GO TO 105
000160          IF(R+HDR.GE.RCARD) GO TO 110
000162          GO TO 102
000163      105 WRITE(6,16) RCARD
000171          GO TO 100
000173      110 WRITE(18) DATA
000201          NPTS=NPTS+1
000203          RTEMP=R
000204          ISORT=1
000205          GO TO 102
000207      112 WRITE(18) DATA
000215          NPTS=NPTS+1
000217          GO TO 102
                C
                C       READ PLOT PARAMETER CARD
                C
000221      115 REWIND 18
000223          READ(5,2) SYM,RCARD,ITH,ILOGO,DEC,TS,ILOGT,DECT,ILINE,SAB,SOR,IPR
000257          IF(SYM.EQ.ANEWR) GO TO 101
                C
                C       CHECK INPUT, IF 0., SET STANDARD VALUES
                C
000263          IF(DEC.LE.0.) DEC=4.
000265          IF(SAB.LE.0.) SAB=0.
000270          IF(SOR.LE.0.) SOR=0.5
000273          IF(TS.LE.0.) TS=1.E-6
000276          IF(DECT.LE.0.0) DECT=2.
                C
                C       INITALIZE PLOT COORDINATES
                C
000301          SPACL=SAB+2.5
000303          XRNG=SAB-.7
000305          IF(RCARD.GE.1000.) XRNG=XRNG-0.1
000311          XTMP=SAB-.28
```

```
000313          XHEAD=SAB-3.25
000315          YHEAD=SOR-.17
000317          ZTH=ITH
000321          IRC=RCARD
                C
                C       INITIALIZE INDICATORS AND COUNTERS
                C
000323          JJ=0
                C
                C       IF SYM = VAY(9) , JOB IS FINISHED
                C
000324          IF(SYM.NE.VAY(9)) GO TO 116
000326          WRITE(6,5)
000331          RETURN
                C
000332        116 WRITE(6,4) SYM,RCARD,ITH,ILOGO,DEC,TS,ILOGI,DECT,ILINE,SAB,SOR
                C
                C       DIAGNOSTICS FOR CARD INPUT
                C
000364          IF(IRC.GT.0) GO TO 117
000370          WRITE(6,3)
000373          GO TO 115
000375        117 IF (ITH.GT.0) GO TO 119
000400          WRITE(6,4)
000403          GO TO 115
                C
                C       DETERMINE ROW AND COLUMN INDICATORS FOR DATA EDITING
                C
000405        119 IRE=ITH/2+1
000410          IRO=ITH/2+2
000412          IRT=1
000413          DO 13  I=1,8
000415          IC=I
000416          YLAB=TAB(IC)
000420          IF(SYM.EQ.VAY(I)) GO TO 132
000422        130 CONTINUE
000424          WRITE(6,5)
000427          GO TO 115
000431        132 GO TO (140,175,163,145,165,150,155,150),IC
000445        140 IC=8
000446          ITIME=2
000447          IR=IRO
000451          GO TO 170
000451        145 IR=IRO
000453          ITIME=IC
000454          IF (LOND .E.LMAX) GO TO 170
000460          IF (ITH.NE.LMAXP1) GO TO 170
000462          IR=1
000463          IC=8
000464          GO TO 170
000464        150 IC=1
000465          ITIME=7
000466          IR=IRE+1
000470          GO TO 170
000471        155 IR=IRE
000473          ITIME=IC
000474          GO TO 170
000475        160 IR=IRO
```

```
000477          IC=9
000500          ITIME=5
000501          GO TO 170
000501      163 IR=IRO
000503          ITIME=3
000504          GO TO 170
000504      165 IR=IRO
000506          ITIME=IC
000507          IF (LGND.GT.LMAX) GO TO 170
000513          IF (ITH.NE.LMAXP1) GO TO 170
000515          IR=1
000516          IC=9
000517          GO TO 170
000517      175 IR=IRO
000521          ITIME=2
                C
                C       EDIT THE DATA
                C
000522      170 ISET1=1
000523          ISET2=1
000524          WRITE (6,22) ICONS,LMAXP2,NPACK,IREM
000540          IF (IREM.NE.0) GO TO 180
000543          LOCV=MOD(IR,2)
000547          IF (LOCV.EQ.0) ISET2=2
000551          IR=IR-1
000553          IRT=IRT-1
000554          GO TO 185
000554      180 LOCT=MOD(IRT+ITIME,2)
000563          LOCV=MOD(IR+IC,2)
000567          IF (LOCT.NE.0) ISET1=2
000571          IF (LOCV.NE.0) ISET2=2
000573      185 IUPACT=(IRT+ITIME*LMAXP2)/2-ICONS
000600          IUPACV=(IR+IC*LMAXP2)/2-ICONS
000604          DO 200 L=1,NPTS
000606          READ(16) DATA
000614          CALL UNPACK(DATA(IUPACT),WORD(1),WORD(2))
000622          T=WORD(ISET1)
000626          IF(T.LT.TS) T=TS
000631          IF(JJ.EQ.1000) GO TO 250
000633          JJ=JJ+1
000634          CALL UNPACK(DATA(IUPACV),WORD(1),WORD(2))
000641          ORD(JJ)=WORD(ISET2)
000645          TM(JJ)=T
000647          IF (JJ.LE.10) PRINT 9,DATA(2),ORD(JJ),TM(JJ)
000672          IF(IPK.NE.12HPK) GO TO 200
000674          WRITE(6,15) JJ
000702          WRITE(6,9) ORD(JJ),TM(JJ)
000717      200 CONTINUE
                C
                C       IF JTYPE = JTHETA, ONLY MAX VALUE PLOTS ARE MADE
                C
000723      250 IF(JTYPE.EQ.JTHETA) GO TO 650
                C
000725          IPLOT=1
000726          IF(ILOGT.NE.LOGT) GO TO 300
000730          IF(ILOGO.NE.LOGO) IPLOT=3
000733          GO TO 400
000734      300 IPLOT=2
```

```
000735          IF(ILOGO.NE.LOGO) IPLOT=4
000740      400 CALL GRAPH(IPLOT,TMP,ORD,XLAB,YLAB,HEADR,SAB,SOR,DECT,DEC,ILINE,IPR
                1 ,ITP,IHC,JJ)
000757      500 IF (JJTYPE.EQ.JTIME) GO TO 115
C
C               SECTION FOR GENERATING MAX VALUE VS. THETA PLOTS
C
000763      650 IF(SYM.NE.VAY(4).AND.SYM.NE.VAY(5).AND.SYM.NE.VAY(7)) GO TO 115
000776          ITP=ITP+1
001000          YMAX=ABS(ORD(1))
001002          DO 655 I=2,JJ
001003          IF(ABS(ORD(I)).GT.YMAX) YMAX=ABS(ORD(I))
001011      655 CONTINUE
001014          ORDTH(ITP)=YMAX
001016          IF(IL.EQ.7) GO TO 660
001020          IF(ITP.NE.ITHETA) GO TO 115
001022          CALL THEPLT(ORDTH,THETA,SOR,SAB,ITP,YLAB,RCARD,HEADR)
001031          GO TO 665
001033      660 IF(ITP.NE.IMTHET) GO TO 115
001035          CALL THEPLT(ORDTH,MTHETA,SOR,SAB,ITP,YLAB,RCARD,HEADR)
001045      665 CALL PLOT(SPACE(0.,*-3)
001050          ITP=0
001051          WRITE(6,7) SYM
C
C               RETURN FOR MORE DATA
C
001057          GO TO 115
001061          END
```